PRESERVING PRIVACY IN WIRELESS NETWORKS

By

Taojun Wu

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Computer Science

August, 2007

Nashville, Tennessee

Approved:

Professor Yuan Xue

Professor Lawrence W. Dowdy

ACKNOWLEDGMENTS

This work was supported in part by TRUST (The Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: Cisco, ESCHER, HP, IBM, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia and United Technologies.

I am especially indebted to Dr. Yuan Xue and Dr. Yi Cui for their guidance, help, and patience with me. Their constant encouragement was instrumental in keeping me motivated in the right direction. I value their systematic academic training, which helped sharpening my critical thinking. More importantly, their devoted mentoring was essential for my graduate study and the completion of this thesis.

Dr. Gautam Biswas, Dr. Larry Dowdy, and Dr. Jerry Spinrad, and other faculty members of EECS, Vanderbilt University, deserve my thanks for equipping me with advanced methodology and supporting my past and future endeavors.

The members of VANETS and TRUST were sources of infinite help and entertainment. Thanks to Bin, Liang, Yann, Nathan, and Jan.

Finally, I would like to thank my family for their support and encouragement.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS ii

LIST OF FIGURES v

Chapter

I. INTRODUCTION 1

II. DIGITAL RIGHTS MANAGEMENT FOR WIRELESS NETWORKS 5

- Introduction 5
- Overall DRM Framework 7
- Composite Wireless Networks Content Decomposition 8
- Hierarchical Key Management and Data Encryption 10
  - Key Generation 10
  - Encryption 12
  - Access Control 12
- Legal EU-Sensor Content Association 13
- Experimental Results 16
  - Testbed Setup 16
  - Evaluation Results 17
- Related Work 18
- Conclusion 19

III. PRIVACY PRESERVATION IN WIRELESS MESH NETWORKS 20

- Introduction 20
- Privacy Preserving Architecture 23
- Privacy Modelling in WMN 25
  - Network Model 25
  - Traffic Entropy 26
- Penalty-based Routing Algorithm 29
- Experimental Results 31
  - Simulation Setup 31
  - Traffic Entropy and Mutual Information 34
  - Which Nodes have more Mutual Information? 35
  - Trade-off between Performance Degradation and Traffic Privacy 37
- Collusion Analysis 39
  - Problem Description 40
  - Colluded Traffic Mutual Information 41

Simulation Results 43
Related Work 47
Conclusion 49

IV. PRIVACY PRESERVATION IN WIRELESS SENSOR NETWORKS 50

Introduction 50
Model 52
Sensor Network Model 52
Attacker Model 53
Privacy Model 57
Optimal Location Privacy Preservation 58
Location Privacy Preservation Algorithms and Simulation 59
Simulation Results 62
Related Work 63
Conclusion 64

V. CONCLUSIONS AND FUTURE WORK 65

BIBLIOGRAPHY 66

LIST OF FIGURES

Figure Page

1. Content Service Architecture in Emerging Composite Wireless Networks. 2
2. Privacy Challenges in Emerging Composite Wireless Networks. 3
3. Digital Rights Management in Emerging Composite Wireless Networks 8
4. Sensor Content Decomposition 9
5. Hierarchical Key Generation 11
6. Label-Guided Content Servicing 15
7. Comparison of Original, Watermarked, Encrypted and Decrypted Images 17
8. Watermarking Time Cost with Different Message Sizes. 17
9. Privacy Preserving Architecture for Wireless Mesh Network. 23
10. An Example of Isomorphic Traffic 25
11. Sampling-based Traffic Analysis 26
12. Experimental Topology 33
13. Traffic Entropy along Time (Single Observer, $\gamma = 1.85$) 34
14. Traffic Entropy in Different Sampling Periods (Multiple Observers, $\gamma = 1.85$) . . 35
15. Sorted Traffic Mutual Information 36
16. Power-law Correlation of Mutual Information and Amount of Traffic Relayed . . 36
17. Average Hop Ratio 37
18. Traffic Mutual Information under Different Penalty Parameters (Destination: Node 1) 38
19. Traffic Mutual Information under Different Penalty Parameters (Destination: Node 16) 39
20. Sorted Traffic Mutual Information under Different Penalty Parameters 40

21. Collusion Reveals Significant Portion of Original Traffic Pattern. 41
22. $I(Y^X, Z^X; X), H(Y^X, Z^X)$ and $H(Y^X, Z^X, X)$ in Venn Diagram. 42
23. Sampled Traffic Curves from Experiment. 43
24. Colluded Traffic Mutual Information (Destination: 1, $\gamma = 1.85$). 45
25. Colluded Traffic Mutual Information (Destination: 16, $\gamma = 1.85$). 45
26. Colluded Traffic Mutual Information (Multiple Pairs of Observers, $\gamma = 1.85$). . . 46
27. Colluded Traffic Mutual Information (Multiple Pairs of Observers, $\gamma = 1.85$). . . 46
28. Colluded Traffic Mutual Information (Multiple Pairs of Observers, $\gamma = 1.85$). . . 47

29. Example Sensor Network. 53
30. Example Penalty Functions of Different Sensitivity to Event Location Revelation. 54
31. Illustration of Directional Traffic Analysis. 58
32. Routing Angle and Guess Angle in Directed Random Walk Routing. 59
33. Privacy Index at Event Source Node. 62
34. Overall Network Privacy Index under Routing Angle and Guess Angle. 63

CHAPTER I

INTRODUCTION

The continued rapid wireless technology advancement has triggered increasing popularity of wireless communications. More types of devices are equipped with wireless functionalities, ranging from embedded sensors, and handheld mobile devices, to stationary routers and desktop PCs. These wireless-enabled devices are then interconnected to form new wireless networks and are further deployed to fulfill varied purposes and demands. Wireless networks are favorable due to the attractive features of relatively low cost, supported mobility, easy deployment with minimal construction, and less reliance on infrastructural facilities.

The wide availability and desirable features of wireless technologies are facilitating many existing tasks with wireless communications and enabling new functionalities. Some of the example scenarios where wireless networks are deployed include: in-home health care, security surveillance, environment monitoring, and Internet connection. Based on different applications and network configurations, the deployed wireless networks serve as information sources (wireless sensor networks, WSN) and information gateways (wireless mesh networks, WMN). Such wide wireless network deployment provides emerging composite networks permeated by wireless communications. The current Internet will remain and act as the "communication bus" in the emerging composite wireless networks. Different wireless sensor networks are deployed to collect raw data from widely spread locations. The collected data is then fed to the Internet which facilitates data distribution. End users acquire desired information by connecting to wireless mesh networks which provides easy Internet access.

The content servicing architecture of emerging composite wireless networks is shown in Fig. 1. We can identify the following three roles in the architecture.

- *Content Provider (CP)* deploys video sensor networks and collects raw data from all these sensors.

- *End User (EU)* requests a subset of sensor content according to individual interest.
- *Service Provider (SP)* acts as interface between *CP* and *EU*. *SP* processes the heterogeneous sets of raw data from different *CP*s, decomposes and transforms them into sensor content with a unified format. The sensor content is customized according to users' requests. When an *EU*'s request arrives, the *SP* parses it and responds back with a customized content subset.

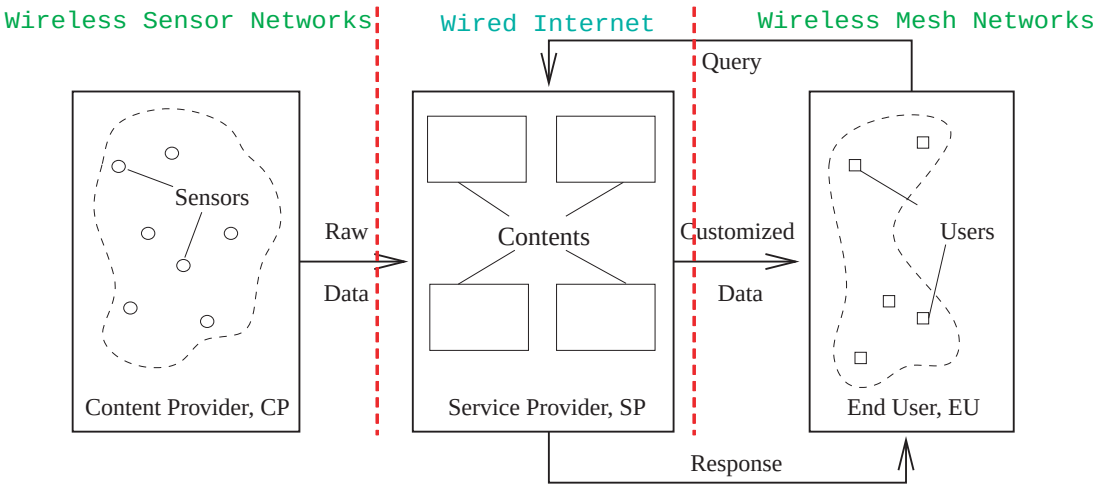


Figure 1: Content Service Architecture in Emerging Composite Wireless Networks.

The emerging composite wireless networks, as we can foresee, involve private personal data in many aspects. Personal information is sensed and collected at many locations (e.g., home, car, work), via many devices. The gathered private data is then transmitted and stored remotely. Furthermore, end users access the collected information with different privileges, depending on his relationship to the data requested. The ubiquitously available data in the networks is vulnerable to privacy threats. Fig. 2 illustrates the privacy challenges that exist in emerging composite wireless networks. How to preserve data privacy at this scale poses a big challenge. This thesis serves as a starting point to the problem.

Data privacy can be content-wise or contextual, depending on how information is obtained from attacker observations. Simply speaking, the content-wise privacy relates to how to answer the question of "what is the information?". For small scale information exchange, many classical

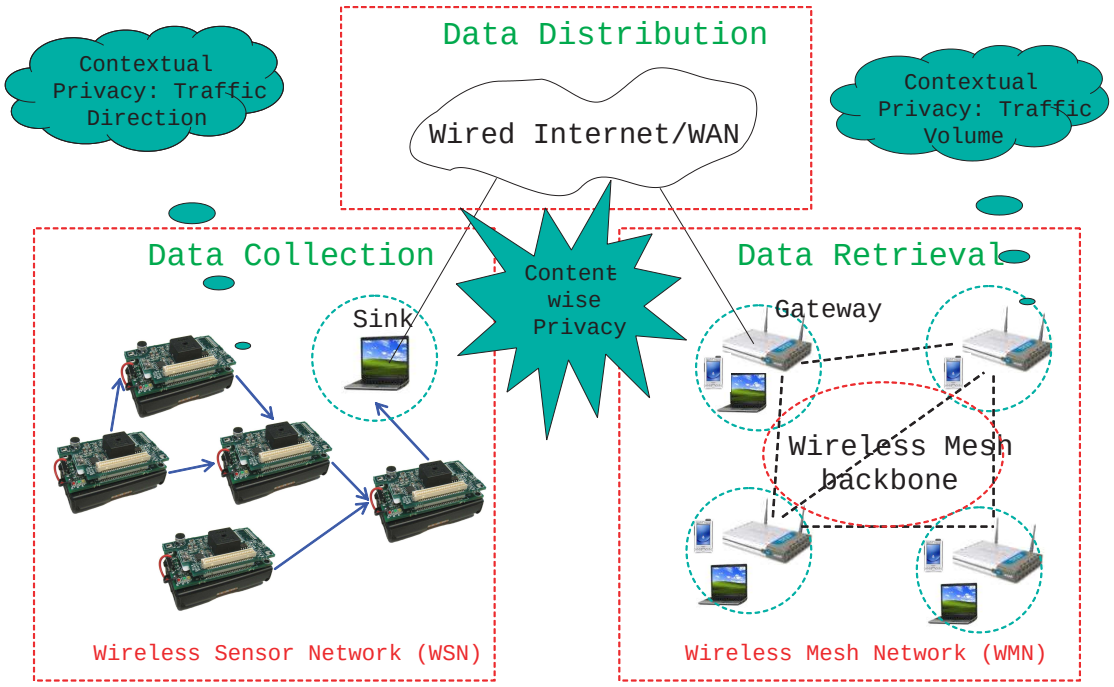


Figure 2: Privacy Challenges in Emerging Composite Wireless Networks.

security protection approaches exist, such as encryption, authentication and access control. When it comes to the case of large-scale massive information flow, the problem becomes hard. This is so because current tools do not scale well and we lack a coherent scheme to handle diversified information flow demands. Our proposal to this situation is to enable customized affordable Digital Rights Management. Chapter II explores to enhance current Digital Rights Management (DRM) schemes with hierarchical key generation to preserve content-wise privacy.

In contrast, contextual privacy relates to the extra information that can be inferred from observations of communication patterns. An attacker who is interested in communication patterns will observe the amount and direction of traffic. In other words, through malicious observation, the attacker will seek answers to questions like “how much information is in transmission?” and “where is the information coming from and going to?”. By doing so, he tries to infer extra contextual privacy information about the traffic. Such threats to contextual privacy are more problematic for wireless networks because of the wireless broadcast communication nature.

In wired Internet, schemes like anonymous routing [51, 33] exist to preserve contextual privacy. In wireless mesh networks and wireless sensor networks, however, contextual privacy is relatively

new research. The relatively high traffic volume in wireless mesh networks makes it vulnerable to volume-based traffic analysis. In Chapter III, this thesis proposes information-theoretic metric *Traffic Entropy* to quantify volume-based traffic analysis. Moreover, it proposes routing control (Penalty-based Shortest Path Routing) to route traffic through diversified random paths to address it. Wireless sensor networks, which are "sense & aggregate" event-driven systems, however, are subject to directional estimation of event sources. This thesis introduces *Privacy Index* to evaluate privacy preservation effect and adopts an optimization-based routing protocol design to find the optimal routing angle for directed random walk routing in Chapter IV. The thesis concludes and points out possible future work directions in Chapter V.

CHAPTER II

DIGITAL RIGHTS MANAGEMENT FOR WIRELESS NETWORKS

Wireless sensor networks are evolving from isolated systems to an integral component of the global information infrastructure, where emerging composite wireless networks serve as the networking component. When sensor networks become public information sources in emerging composite wireless networks and the Internet provides easy information access to numerous end users, DRM (Digital Rights Management) must be enforced, due to the sensitivity and the privacy nature of sensor content. Moreover, existing DRM solutions do not suffice, because the explicit one-to-one mapping between content producer and consumer does not apply in the composite wireless networks. In this chapter, a DRM-enabled content service architecture is proposed. For ease of description, we use video sensor network as an example WSN when explaining the DRM scheme. Within this architecture, we propose a binary-tree-based hierarchical key generation scheme for data encryption, and adopt a label-guided watermarking strategy to enable content abuse trace-back.

Introduction

Sensor networks have dramatically changed the way people interact with the physical world. They are deployed in a physical field collaborating to perform tasks from collecting information such as temperatures and real-time video images to locating the positions of tracking objects. In video sensor networks [16, 39, 35], each sensor is equipped with a camera which can provide important visual information. The content collected by sensor systems not only holds practical value to individuals running them, but also can potentially benefit many other users. For example, a video sensor system monitoring the garage of a shopping mall is setup for security purposes. However, the archived video footage can become valuable material for studies on customer shopping behaviors.

When sensor network becomes a public information source on the Internet, many urgent technical issues arise, mainly due to the sensitivity and the privacy nature associated with the sensor content. In this chapter, we argue for the necessity of enforcing DRM (Digital Rights Management) of content servicing in emerging composite wireless networks. Here, DRM refers to a collection of technologies used to handle the description, valuation, trading and monitoring of the rights held over any digital entity. DRM has been proved technically sound in protecting digital work copyrights in movie and music industries [53]. Mature DRM systems have also been developed [38, 9]. However, many intrinsic difficulties arise when deploying existing DRM solutions into emerging composite wireless networks.

The challenge comes from the distinguishing data characteristics of the traditional digital content and sensor content. In typical DRM applications, an explicit one-to-one mapping exists between the producer and the consumer of the digital content, such as movies and music titles. Essentially a binary file, each piece of content is encrypted by a unique secret key prepared by its producer (i.e., the owner and distributor). End users, as the content consumer, must purchase a license that contains this key, in order to enjoy the content. Furthermore, the user's access to a piece of content is all-or-nothing (e.g., an interested user must gain access to the movie in its entirety and not any of its subsets).

Such a one-to-one mapping vanishes in the domain of sensor networks. First, the sensor content is the spatial and temporal composition of data inputs from all sensors in the network. With respect to the information provided, the data streams produced by different sensors are often co-dependent. From the viewpoint of end users, what a meaningful piece of content (e.g., temperature in the playground) embodies is clearly detached from how it is produced (i.e., which sensors collectively created this result). Second, a user's view towards the sensor content is often partial and customized due to factors like user interest and privacy protection. For example, in home monitoring for patient care, a video sensor network collects footage of patients within a geographical region. The caretaker of a patient may choose to view his/her activity during a certain period of time, but is clearly forbidden to view the footage of other patients within the same network.

In light of these challenges, any DRM solution for the sensor network must have extreme built-in flexibility during the collection, preparation, and access of sensor content. Furthermore, the DRM solution needs to effectively balance the trade-off among flexibility for content management, management overhead for content servicing, and usability for end users. We propose a DRM-enabled content service architecture for emerging composite wireless networks. The three essential parts of this framework are:

- Provide content decomposition for data streams to enable flexible data retrieval.
- Introduce binary-tree-based hierarchical key generation to support scalability for large-scale communication demands in continuously growing networks.
- Service customized contents following unique labels for every request. This will allow the Service Provider to locate the malicious user when a content breach occurs.

The rest of this chapter is organized as follows. Sec. II describes the major entities in the content service architecture and presents the digital right management framework. Sec. II illustrates how content is decomposed, using video sensor network as an example. Sec. II details the security components in the DRM framework. Sec. II introduces the label-guided watermarking scheme to discourage content abuse and help locate violators. Sec. II gives an overview of related work in digital rights management. Our evaluation results obtained from preliminary testbed system are presented in Sec. II. We provide conclusions in Sec. II.

Overall DRM Framework

Integrating the three essential parts together, the overall DRM framework is shown in Fig. 3. The five important components of DRM that are implemented by the *SP* are: Content Server, Query Server, Policy Server and License Server. The Query Server parses *EU*'s queries. The Policy Server is responsible for sensor data content access control. The License Server tracks all past and present encryption keys. It also handles license requests from the *EU* and records

granted access rights of individual users. The Content Server stores watermarked and encrypted content units and dispatches requested contents to the *EU* in designated ways. The Encryption/SP Watermarking component provides security functionality like encryption, message authentication, digital signature, license generation, and label-based watermarking. On the *EU* side, the DRM Manager handles the requesting and verifying of a license and enforcing a digital rights check. The Decryption component decrypts contents for the Content Player to feed playback to *EU*. The *CP* owns deployed sensors and collects raw data from those sensors and supplies to to *SP*.

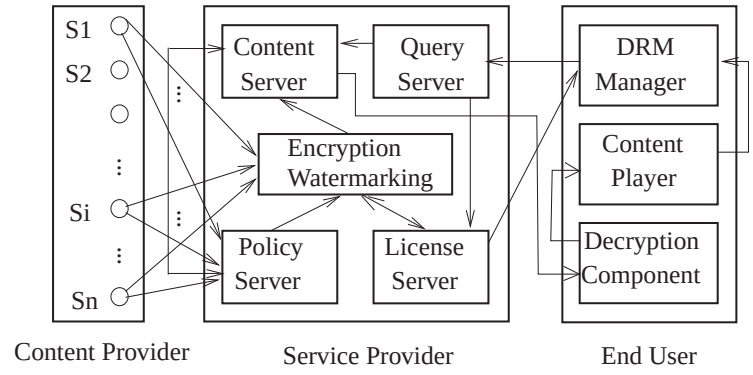


Figure 3: Digital Rights Management in Emerging Composite Wireless Networks

Our framework is viable under several attack scenarios. For typical eavesdropping attacks, even if an attacker can intercept the video contents sent to an *EU* A, he can not possibly acquire all necessary keys to decrypt them. Furthermore, illegal distribution is countered as well. Any valid *EU* A is identified by a unique $UserID_A$, which relates to his *binary identification key*. In case of a content breach by B, the breach string will identify him. Sec. II explains these aspects in more detail.

Composite Wireless Networks Content Decomposition

From Fig. 1, we identify three entities, *CP*, *SP* and *EU* in emerging composite wireless networks content servicing architecture. In this section, we present a decomposition of sensor content in emerging composite wireless networks and how this is collectively done by the three entities. The flow of sensor content, from its creation and collection, to its distribution, is outlined in Fig. 1.

Defining an interface to map between the raw sensor data and the content customizable by the end user is the main purpose to implement this service architecture. The main challenge, as described in the introduction, comes from the reality that the explicit one-to-one mapping between the content producer and consumer in traditional DRM applications does not apply for the domain of sensor networks.

We start with the raw image data/video data collected by the video sensors. Let S_i ($i = 1, 2, ..., N$) be one of the N video sensors in the network. The data content is provided by S_i as a content stream $ConStream_i$ across a three dimensional domain (one dimension in temporal domain and two dimensions in spatial domain). In the temporal domain t_i, $ConStream_i$ consists of a series of content items ($ConItem_i(t), t \in [t_i^1, t_i^2]$). In the spatial domain, each $ConItem_i(t)$ (or video frame) is decomposed into small content units ($ConUnit_i^j(t), j = 1, 2, \ldots$), which are the smallest content units to respond to user-specific queries. A $ConUnit$ is part of a video frame and is the smallest element for encryption at the content server of CP. A choice of encryption at this granularity serves two purposes: to save only useful information and to support customization of *EU*s. The relationship of these three units is illustrated in Fig. 4. Mathematically, we have $ConUnit_i^j(t) \in ConItem_i(t)$, $ConItem_i(t) \in ConStream_i$ and $\bigcup_{i \in [1,N]} ConStream_i$ constitutes the whole set of content provided by CP.

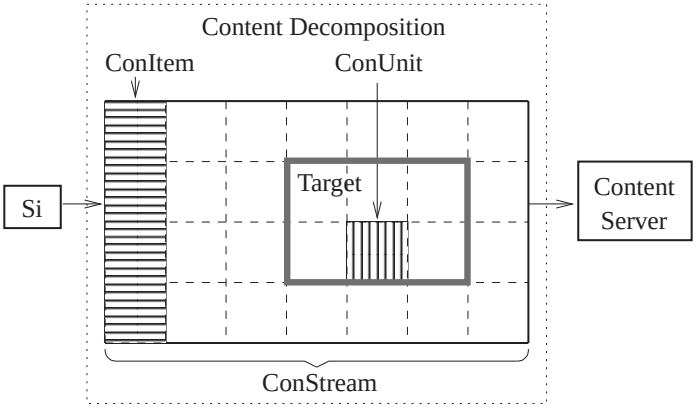


Figure 4: Sensor Content Decomposition

From a user's perspective, the unit of interested content is defined as a *target*. In a video sensor network, a target may be a home, a highway, a garage, etc. When an *EU* requests the desired target from a sensor content service, he/she will submit a profile of the target which may include its identity, position, size, and time interval of the request. Based on the profile, the *SP* maps the target into a set of content units, which collectively embody this target. The set of content units are then delivered to the *EU*.

For example, a target's profile can be denoted as $\{ID =$"BankFrontDoor", $POS = (5, 7)$, $SIZE = (9, 15), TIME = [100s, 120s]\}$. This maps to a set of images $Img_i(t)$ with region size 9 by 15 (in pixels) at position (5,7) during time t, $100s \leq t \leq 120s$.

Hierarchical Key Management and Data Encryption

When accessing customized content from emerging composite wireless networks, the user requests can be highly heterogeneous across spatial and temporal domains, at varying granularity. In our content service architecture, some requests may only involve a handful of decomposed content units, while others may cover thousands of them. Obviously, it is not realistic to find a one-size-fits-all solution by looking for the right size of the content unit. To address this challenge, we present a *tree based hierarchical key management and encryption scheme* for data encryption in this section.

Key Generation

The design goal of key management at the *SP* is to support a scalable data encryption solution that is adaptive to highly heterogeneous sensor content requests. The basic idea is to generate a hierarchical key structure corresponding to the content item structure. The keys at the lower level of the hierarchy could be generated from the keys at the higher level. For each content unit ($ConUnit$ – the smallest unit that corresponds to users' requests), the keys at the lowest level (leaf keys) are used for encryption. When a content stream ($ConStream$) that consists of multiple $ConUnit$s is requested, instead of providing all the leaf keys that encrypt these $ConUnit$s to the end user, only the keys which constitute the minimum cover of these keys in the tree hierarchy are provided.

Specifically, the *CP* and *SP* first reach an agreement about $MasterKey_P$, a common provider's master key. This $MasterKey_P$ is the highest-level key in the hierarchy and is used throughout their sensor network content provision contract. For every sensor S_i, *SP* generates a sensor master key $MasterKey_i$ using a hash function with S_i's profile and the provider's master key as input. All $MasterKey_i$s are updated on a regular basis.

$$MasterKey_i = HASH(SensorProfile_i || MasterKey_P) \tag{1}$$

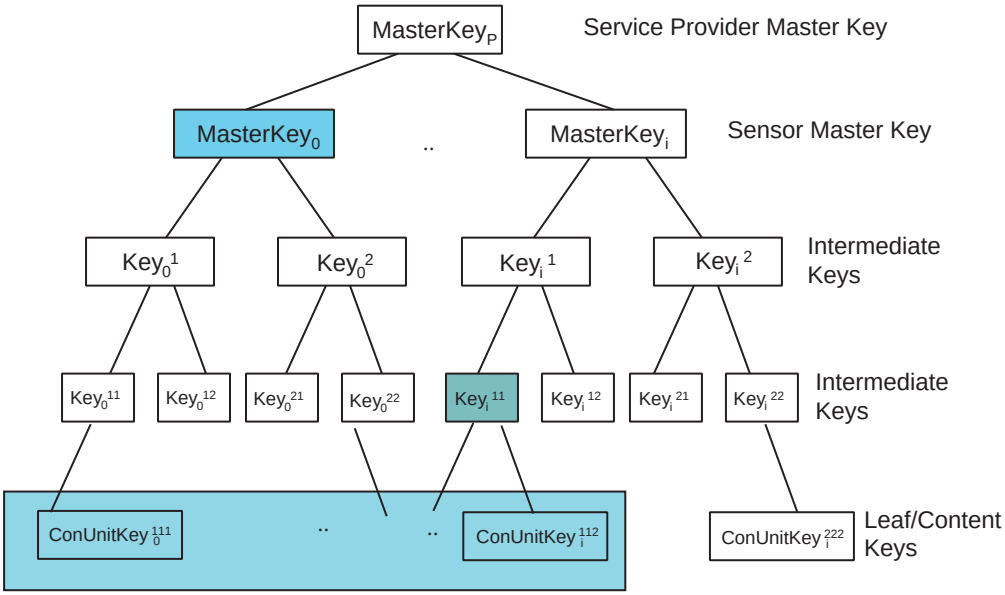


Figure 5: Hierarchical Key Generation

For each sensor S_i, its $MasterKey_i$ is used to generate content unit keys ($ConUnitKey_i^j, j = 1, 2, \ldots$) through a tree-based key hierarchy as shown in Fig. 5. At each level k, we associate intermediate keys with the root node $Key_i^0 = MasterKey_i$. The leaf nodes provide content unit keys $ConUnitKey_i^j$.

$$Key_i^{(k-1)||l} = HASH(1||Key_i^{k-1}) \tag{2}$$

Here, l represents the "tree node position" which could be a time range, or the region position of the image. At time t, the content unit ($ConUnit_i^j(t)$) will be encrypted with the content unit

key ($ConUnitKey_i^j$) corresponding to the leaf node. The License Server in *SP* keeps a log of all master, intermediate and content keys ever generated and used within the DRM system.

Encryption

Note that when the sensor content moves from *CP* to *SP*, they are encrypted by link-level protocols like TinySec. At the *SP*, sensor content is decomposed into content units and further encrypted by its corresponding content keys with conventional symmetric ciphers. Upon end users' requests, content units of targets resulting from decomposition of the *EU* request will be delivered encrypted.

The same $ConUnit$s of $ConStream_i$ from one sensor S_i will be encrypted with one single key for a certain period, as described in the key generation step. Hence, a potential adversary possessing access privileges to one target content cannot decipher other unlawfully acquired target contents, even if they are all from the same sensor source. The periodically updating characteristic of encryption key of the same target ensures that a user is granted only limited time length access rights to his desired target. To be able to access content over a larger time domain, the *EU* needs to request appropriate access rights through multiple licenses.

Access Control

Digital right management enables the sensor data content to be delivered to end users via diverse, non-secure communication channels. To actually playback the received encrypted contents, the *EU* will need to authenticate himself and request appropriate access rights for the sensor contents. To do so, the *EU* will request a license from the *SP*. In his request, the *EU* will indicate for what content and time interval he is requesting a license. Also included will be his personal information (e.g. unique ID or credit card number, if payment is necessary).

After user authentication, the *SP* will verify his access rights to the requested sensor content units and generate a license. The license includes all the keys (intermediate and content keys as will be detailed soon) required to decrypt the contents. In our tree-based key hierarchy, this corresponds to a minimum cover of all the leaf keys that correspond to the content units requested

by the user. As illustrated in Fig. 5, if the *EU* requests all content units as marked in a rectangle, then instead of returning all keys associated with each unit, the *SP* only needs to return the keys of the shadowed nodes, which are $MasterKey_0$ and Key_i^{11}. From the key generation procedure, it is obvious that the user is able to derive all content keys required to decrypt the content he requested. To protect the confidentiality and integrity of the license, the license is signed with the *SP*'s private key from the License Server and encrypted via the *EU*'s public key.

Our hierarchical key management solution scales well to user requests for large volume contents. It reduces the overhead and complexity involved in communicating the keys to the end users. At the same time it is also flexible enough to meet the diverse user requests for sensor content with different sizes (i.e., number of content units). For a binary hierarchical key tree, the key hierarchy doubles the key space that the *SP* needs to manage, in comparison with a flat key management solution where content unit keys are organized in a flat way. This is so because in a complete binary tree the number of leaf nodes is half that of the total number of nodes. Yet as *SP* usually resides on powerful servers, such increase in key space would not significantly affect the performance of our DRM framework for video sensor content service.

Legal EU-Sensor Content Association

Watermarking [42, 19] is the process of embedding data into a multimedia content such as image, audio and video. The embedded information, called a watermark, can be extracted later on for security reasons. In our DRM framework, digital watermarking of the generated sensor content at the *CP* and the *SP* is used to (1) protect the rightful ownership of the *CP* and the *SP*; (2) discourage the *EU*s from abusing their digital rights and enable the *CP* and the *SP* to trace illegal sensor content distributions and identify violators.

In particular, the *SP* prepares a composed sensor content consisting of individual sensor content units, desired and requested by the *EU*. First, the *CP* and the *SP* generates DRM-safe sensor content, where each sensor content unit carries the hierarchical watermark consisting of *CP and SP* rightful ownership information. However, this composed watermark is not sufficient to create

a sensor content stream that would be unambiguous for different *EU*s. To address this issue, we present a third watermarking process laid over sensor content – *label-guided watermarking scheme*, which is able to provide efficient yet powerful digital watermarking for unambiguous and legal sensor content association with an *EU*.

The label-guided watermarking scheme has the following steps: First, the *SP* chooses two watermarks W_0 and W_1. Let $S^j = (S_1^j, ..., S_n^j)$ be the composed sensor content stream prepared for an *EU* j. As the next step, the *SP* copies the stream S^j to create $S^{j'}$ stream and watermarks the sensor content stream S^j with watermark W_0 and stream $S^{j'}$ with watermark W_1. Then the *SP* generates a unique *EU/customer* label in the form of a *binary identification key* b (e.g., 01101010100). Such a label is generated by a hash function based on the information in the user's request (e.g., user's ID, query content, time stamp)

$$b = HASH(UserID||UserQuery||TimeStamp) \tag{3}$$

This generated label string is used to determine which watermark each outgoing content unit should have. Finally, a watermarked composed sensor content stream S_{final}^j which is unique to the *EU* is generated as follows.

- If the identification key binary digit is 0, then a sensor content unit from the stream S^j watermarked with watermark W_0 is selected;
- If the identification key binary digit is 1, content unit from $S^{j'}$ with watermark W_1 is selected.

This implies that the generated label string is able to determine which watermark each outgoing content unit should have[1]. That is, depending on if b_k is 0 or 1, the kth content unit has watermark W_0 or W_1. In this way, the combination of two different watermarks in content units reveals the source (one content service at a particular time to a particular user) of leaked digital contents. The process of label guided content service is shown in Fig. 6. If the number of content units in a

[1]Assume that all $ConUnit_i^j(t)$s are totally ordered.

single request exceeds $|b|$, the label is scanned from left to right repeatedly until all content units are dispatched.

For example, let us assume

$$S = (Content_{S_1}^{W_0}, Content_{S_2}^{W_0}, Content_{S_3}^{W_0}, Content_{S_4}^{W_0}),$$
$$S_{copy} = (Content_{S_1}^{W_1}, Content_{S_2}^{W_1}, Content_{S_3}^{W_1}, Content_{S_4}^{W_1})$$

and the content units are coarsely partitioned at the level of different sensor information (note that the granularity in reality is much finer, going into small $ConUnit_i^j$ units). If the *EU* identification key is 0110, then the resulting sensor content stream for the *EU* is

$$S_{final} = (Content_{S_1}^{W_0}, Content_{S_2}^{W_1}, Content_{S_3}^{W_1}, Content_{S_4}^{W_0}).$$

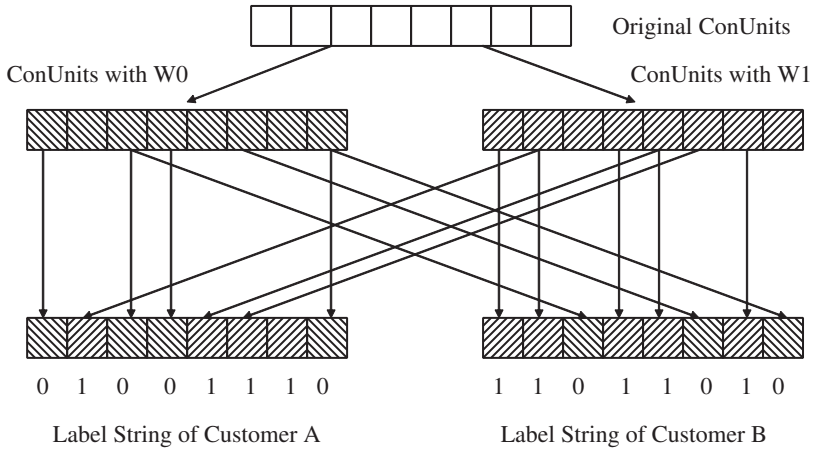


Figure 6: Label-Guided Content Servicing

Utilizing the label-based watermarking we are able to study the attacks, collusion possibilities among *EU*s, and leaking actions. Due to the fact that a single content unit, or a small collection of content units (say, less than $|b|$) are meaningless, and that locating such small-scale actions are extremely difficult, we assume a content breach action to be one that leaks out at least $|b|$ content units. Suppose a subset of m content units are breached, and a series of bits $(0, 1)$ denote

whether each content unit is watermarked with W_0 or W_1. We can arrange these bits into a unique Breach String $BStr$ of binary bits $b_1, b_2, \ldots, b_m$ following their corresponding $ConUnit$s' order. By inspecting $BStr$, we can identify a repeating substring $subBStr$ of length $|b|$. This $subBStr$ will uniquely identify the leaker.

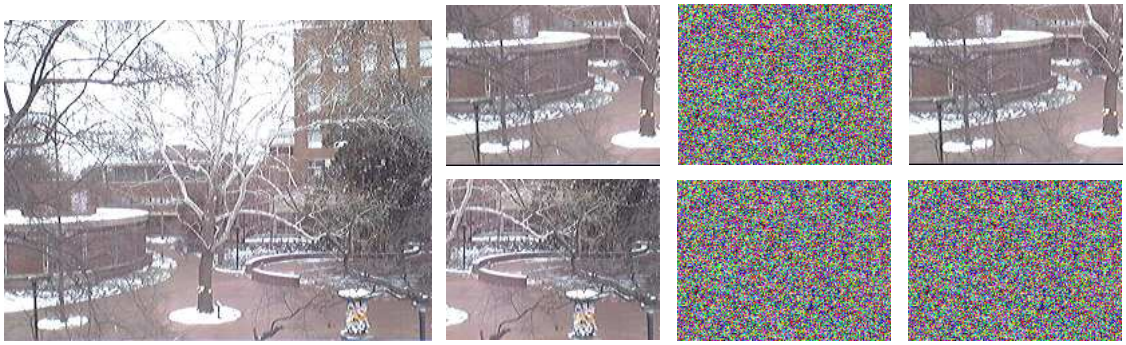
Experimental Results

Testbed Setup

We build our testbed system on a Linux system (Dell Precision 670, dual-core, 2GB RAM). We prepared a set of webcam images (average size 225KB) to simulate the video sensor content. These images have a unified resolution of 320 x 240. Each image is split into several $ImgReg$s, each representing a target. Every $ImgReg$ is duplicated to two copies, watermarked with different keys. A unique encryption key is generated separately for each $ImgReg$. In the preliminary experiments reported in this thesis, we evaluate the average performance of watermarking, encryption, and decryption on each $ImgReg$.

Our watermarking experiment uses the watermarking scheme provided by the Digital Invisible Ink Toolkit (DIIT)[2]. The encryption experiment is built upon the Java security library. Furthermore, we rely on the Message Digest feature to produce a unique encryption key for each target. The encryption algorithms we choose are DES and RC4.

Fig. 7 illustrates the flow of our experiments. Fig. 7 (a) shows a webcam image of the engineering campus of Vanderbilt University. The original image is then split into four $ImgReg$s, with each watermarked individually. Fig. 7 (b) shows the watermark results of the bottom two $ImgReg$s. The same $ImgReg$s after encryption are shown in Fig. 7 (c). Finally, in Fig. 7 (d), we show client-side result of a user interested in solely monitoring the Small Molecule NMR Facility Core, which is the round building shown in the lower left $ImgReg$ of the original image. While this $ImgReg$ is decrypted by acquiring the corresponding decryption key, the remaining $ImgReg$ remains encrypted from the viewer.



(a) Original Image (b) Watermarked Images (c) Encrypted Images (d) Decrypted Images

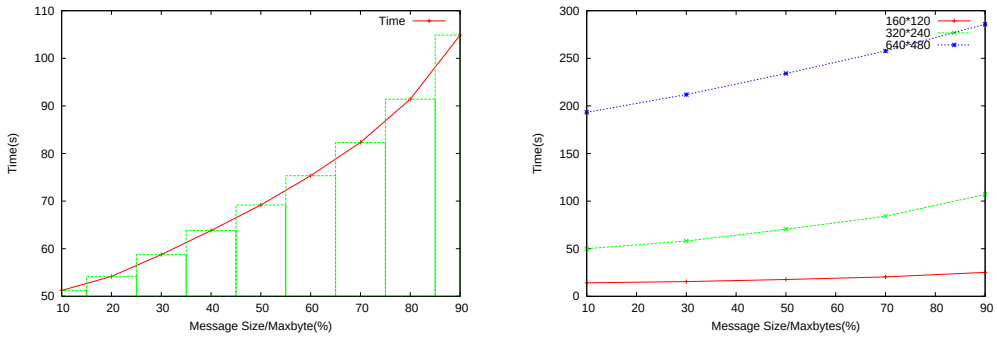
Figure 7: Comparison of Original, Watermarked, Encrypted and Decrypted Images

Evaluation Results

Watermark Size Effect

Fig. 8 (a) demonstrates the time overhead of watermarking on the same image using watermarks of different sizes. For horizontal axis, we show the ratio of watermark size to $MaxBytes$, the maximum bytes that can be hidden in an image. According to DIIT[2]:

$MaxBytes = (ImageHeight \times ImageWidth \times$ Color Numbers $\times$ Number of Bits to Hide$)/8$.



(a) same image (b) different images

Figure 8: Watermarking Time Cost with Different Message Sizes.

As shown in the picture, the time overhead does not grow linearly as a function of the message size. This is because as the ratio becomes larger, it takes the watermarking program longer time to find the free space and hide the information.

Fig. 8 (b) compares results of watermarking two images of different sizes using the same set of watermarks. The smaller image is 6.9KB, and the larger one is 20.8KB. The sizes of messages range from 1KB to 24KB. In the picture, the two curves are almost overlapped. Although the large image does take longer time than the smaller one, the difference is trivial. Combined with the results in Fig. 8 (a), the size of the watermarks has greater impact than the size of the image.

Key Generation Performance

We test the time to create a finite number of keys. The results show that it takes 150ms to create 100 keys but takes only about 450ms to create 10,000 keys. Since we run the key creation function repeatedly, and did not save the created keys into the storage system, the results may be optimistic. But even for 150ms/100 keys, it can still support a system with a large key number requirement.

Related Work

Our work relies on extensive prior work in Digital Rights Management (DRM) in distributed multimedia systems and research results in several areas including watermarking encryption algorithms and security protocols. Major players in Internet-based multimedia have adopted DRM into their mainstream products. Windows Media DRM [9] is a flexible platform that makes it possible to protect and securely deliver content by subscription or by individual request. Developed by RealNetworks, Helix DRM [4] is a comprehensive and flexible platform for the secure media content delivery of standards-based as well as leading Internet formats, including RealAudio, RealVideo, MP3, MPEG-4, AAC, and H.263. Both solutions provide secure media packaging, license generation, and content delivery to a trusted media player on a computer, portable device, or network device. DRM has also been applied in preserving the privacy of user context information in ubiquitous computing environment [25]. However, none of the existing DRM solutions are applicable to protect the video sensor content due to the challenges we have presented. In [50, 41], two hierarchical access control and key management frameworks are presented. Our work is different

from [50, 41] in that we consider the unique temporal and spatial diversity characteristics of video sensor contents.

Security for wireless sensor networks has been extensively studied in the existing literature, which includes link layer security [31], broadcast authentication [44], and key management [24]. Concerned to the security issues involved with the emergence of sensor networks, the existing research has focused on protecting the information within sensor networks. Our work mainly focuses on preserving the privacy and economical value of the sensor information when it is delivered from sensor networks to the Internet.

Conclusion

Digital right management is a critical component to enable the vision of sensor-centric global information infrastructure. This paper presents the architecture and the enabling security mechanisms of digital right management for video sensor networks. Novel key management scheme is presented to address the unique challenge of video sensor data content distribution. Initial testbed results show that our proposed solution is sound and efficient. We will expand our experiment from single images to continuous streams as future work.

CHAPTER III

PRIVACY PRESERVATION IN WIRELESS MESH NETWORKS

Multi-hop wireless mesh networks (WMN) have attracted increasing attention and deployment as a low-cost approach to provide last-mile broadband Internet access. Privacy is a critical issue in WMN, as traffic of an end user is relayed via multiple wireless mesh routers. Due to the unique characteristics of WMN, the existing privacy solutions applied in the Internet are either ineffective at preserving privacy of WMN users, or will cause severe performance degradation.

In this chapter, we propose a light-weight privacy preserving solution aimed to achieve a well-maintained balance between network performance and traffic privacy preservation. At the center of this solution is a novel metric called "traffic entropy", which quantifies the amount of information required to describe the traffic pattern and to characterize the performance of traffic privacy preservation. We further present a penalty-based shortest path routing algorithm that maximally preserves traffic privacy by minimizing the mutual information of "traffic entropy" observed at each individual relaying node, meanwhile controlling performance degradation within the acceptable region. Extensive simulation evidence indicates the soundness of our solution.

Our solution is further tested in the case of collusion of two malicious observers. Simulation results show our approach is resilient to two colluding observers.

Introduction

Recently, multi-hop wireless mesh networks (WMN) are being deployed as a low-cost substitute approach to provide "last-mile" broadband Internet access [5, 7, 8, 6]. In a WMN, each client accesses a stationary wireless mesh router. Multiple mesh routers communicate with one another to form a multi-hop wireless backbone that forwards user traffic to a few gateways connected to the Internet. Some perceived benefits of WMN include enhanced resilience against node failures and channel errors, high data rates, and low costs in deployment and maintenance. For such reasons,

commercial WMNs are already deployed in some US cities (e.g., Medford, Oregon). Even large cities are planning to deploy city-wide WMNs as well [1].

However, to further widen the deployment of WMN, and enable them as competitive players in the market of broadband Internet access, privacy issues must be addressed. Privacy has been a major concern of Internet users [17]. It is a particularly critical issue in the context of WMN-based Internet access, where users' traffic is forwarded via multiple mesh routers. In a community mesh network, this means that the traffic of a residence can be observed by the mesh routers residing at its neighbors. Despite the necessity, limited research has been conducted towards privacy preservation in WMN.

This motivates us to investigate the privacy preserving mechanism in WMN. There are two primary privacy issues – data confidentiality and traffic confidentiality.

- *Data confidentiality*. Data content reveals user privacy on what is communicated. Data confidentiality aims to protect the data content and prevent eavesdropping by intermediate mesh routers. Message encryption is a conventional approach for data confidentiality.

- *Traffic confidentiality*. Traffic information (e.g., who the users are communicating with, when and how frequently they communicate, the amount and the pattern of traffic) also reveals critical privacy information. The broadcasting nature of wireless communication makes acquiring such information easy. In a WMN, attackers can conduct traffic analysis at mesh routers by simply listening to the channels to identify the "ups and downs" of target's traffic. While data confidentiality can be achieved via message encryption, it is harder to preserve traffic confidentiality. In this chapter we focus on the user traffic confidentiality issue, and study the problem of traffic pattern concealment.

We aim at designing a light-weight privacy preserving mechanism for WMN which is able to balance the traffic analysis resistance and the bandwidth cost. Our mechanism makes use of the intrinsic redundancy of WMN, which is able to provide multiple paths for data delivery. Intuitively, if the traffic from the source (*i.e.*, gateway) to the destination (*i.e.*, mesh router) is split among

many paths, then all the relaying nodes [1] along the paths can only observe a portion of the entire traffic. Moreover, if the traffic is split in a random way, both spatially and temporally, then an intermediate node has limited knowledge to figure out the overall traffic pattern. Thus the traffic pattern is concealed.

Based on this intuition, we seek a routing scheme which routes data such that the statistical distributions of the traffic observed at intermediate relaying nodes are independent from the actual traffic from the source to the destination. To achieve this goal, we first define an information-theoretic metric – "*traffic entropy*", which quantifies the amount of information required to describe the traffic pattern. Then we present a penalty-based routing algorithm, which aims to minimize the mutual information of "traffic entropy" observed at each relaying node, meanwhile controlling the network performance degradation to an acceptable level.

Considering the possibility of collusion, we evaluate our scheme under situation when two observers exchange their knowledge about the same destination. We measure this shared knowledge as "colluded traffic mutual information" and our simulation results show that our scheme is still viable in case of two colluding eavesdroppers.

The rest of this chapter is organized as follows. In Section III, we present the overall architecture for privacy preservation in WMN. Sections III and III focus on the traffic privacy issue. In particular, Section III presents a model to quantify the performance of traffic privacy preservation, and Section III presents a routing algorithm. The proposed privacy preserving solution is evaluated via extensive simulation in Section III. Section III discusses possible collusion problems with malicious traffic observers and its impact on our proposed scheme. Section III summarizes background knowledge and related work. Section III concludes the chapter and points out the future directions.

[1] In this thesis, we use the following terms interchangeably: wireless mesh router, intermediate relaying node, wireless node.

Privacy Preserving Architecture

We consider a multi-hop WMN shown in Fig. 9. In this network, client devices access a stationary wireless mesh router at its residence. Multiple mesh routers communicate with one another to form a multi-hop wireless backbone that forwards user traffic to the gateway which is connected to the Internet.

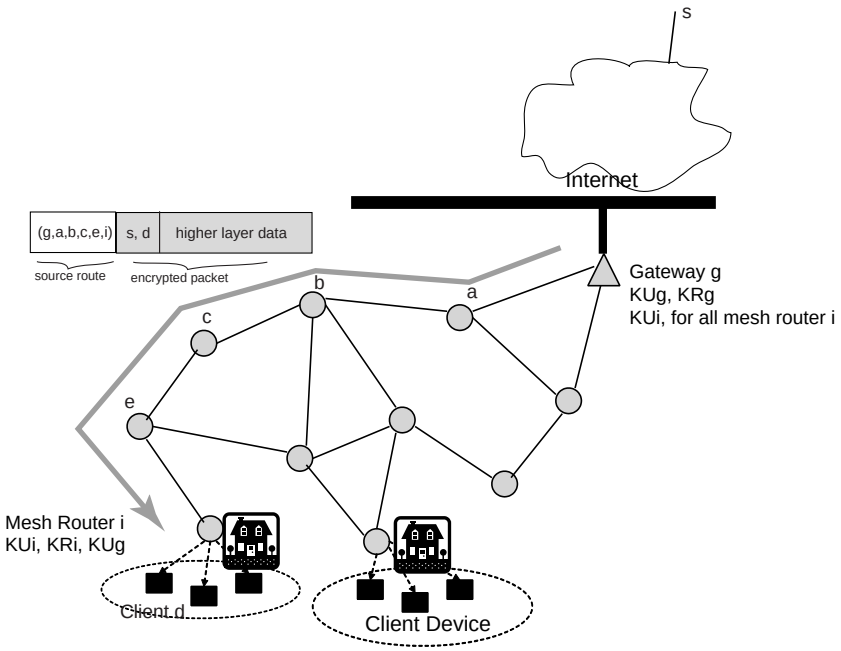


Figure 9: Privacy Preserving Architecture for Wireless Mesh Network.

Two privacy aspects are considered in this architecture. *Data confidentiality* aims to protect the data content from eavesdropping by the intermediate mesh routers. *Traffic confidentiality* prevents a traffic analysis attack from the mesh routers, which aims at deducing the traffic information such as who the user is communicating with, the amount, and the pattern of traffic. Our privacy preserving architecture aims to protect the privacy of each wireless mesh router, the basic routing unit in WMN. The architecture consists of the following functional components.

- *Key Distribution*. In this architecture, each mesh node, as well as the gateway, has a pair of public and private keys (KU, KR). The gateway maintains a directory of certified public keys of all mesh nodes. Each mesh node has a copy of the public key, KU_g,of the gateway.

The public key KU_i of mesh node i and KU_g are used to establish the shared secret session key KS_{gi}, which is used to encrypt the messages between them.

- *Message Encryption.* Let M be the IP packet sent from a source s in the Internet to a client d in the mesh network, and i be the mesh router of client d. The IP packet M, which contains the original source and destination address s and d, is encrypted at gateway g via the shared secret key KS_{gi}: $M_e = E(KS_{gi}, M)$. To route the encrypted packet M_e to its destination, the gateway prefixes to the packet the source route from the gateway g to the router i. The encapsulated packet is then forwarded by relaying routers in WMN. Likewise, packets traveled in the reversed direction are treated the same way. As the source address s and other higher layer header information (e.g., port, ID), are all encrypted, the relaying routers are unable to obtain the information on who the client of router i is communicating with, and what type of application is involved. Since encryption and decryption take place only at the gateway and the destination mesh router, much less computation is required, which is a desired feature in WMN.

- *Routing Control.* With the source route in clear text in an encapsulated packet, the intermediate mesh routers can still observe the amount and the pattern of the traffic of a particular mesh node i. To address this problem, our privacy preserving mechanism explores the path diversity of WMN, and forwards packets between the gateway and the mesh node via different routes. Thus, any relaying router can only observe a portion of the whole traffic of this connection. In Section III, we detail the design of a penalty-based routing algorithm, which randomly selects a route for each individual packet such that the observed traffic pattern at each relaying node is independent of the overall traffic. The residential networks are generally small in size. Therefore, in our design, the gateway maintains a complete topology of the WMN, and computes the source routes between the destination mesh nodes and itself.

Network Model

We model the WMN shown in Fig. 9 as a graph $G = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of wireless nodes in WMN, and $\mathcal{E}$ is the set of wireless edges (x, y) between any two nodes x, y. Each node x maintains a logical connection with the gateway node g. Node x receives data from the Internet via g. The source and destination information of a packet is open to the relaying node. The traffic pattern of x can be categorized into two types: incoming traffic patterns and outgoing traffic patterns. In this paper, we concentrate on the first type.

If the traffic between g and x goes through only one route, then any relaying node on this route can easily observe the entire traffic between g and x, thus violating its traffic pattern privacy. To avoid this problem, x must establish multiple paths with g and distribute its traffic along these paths, such that any node can only reconstruct a partial picture of x's traffic pattern.

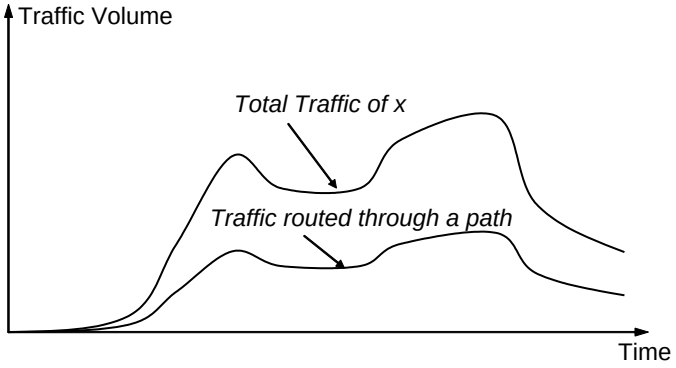


Figure 10: An Example of Isomorphic Traffic

However, the complete traffic pattern information of x could still be obtained by a single node in case of multi-path routing. In the example shown by Fig. 10, g allocates the traffic to x via two disjoint routes by fixed proportion. Then, for any node along any path, although only seeing one half of the flow, the observed traffic shape is isomorphic to the original one. Therefore, the traffic

to x must be distributed along multiple route in a time-variant fashion, such that the traffic pattern observed at any node is statistically different from the original pattern.

Traffic Entropy

We propose to use information entropy as a metric to quantify the performance of a solution at preserving the traffic pattern confidentiality. In what follows, we consider two nodes x and y. x is the destination node of the traffic from the gateway g to x. y is the observing node, which relays packets to x and also tries to analyze the traffic of x.

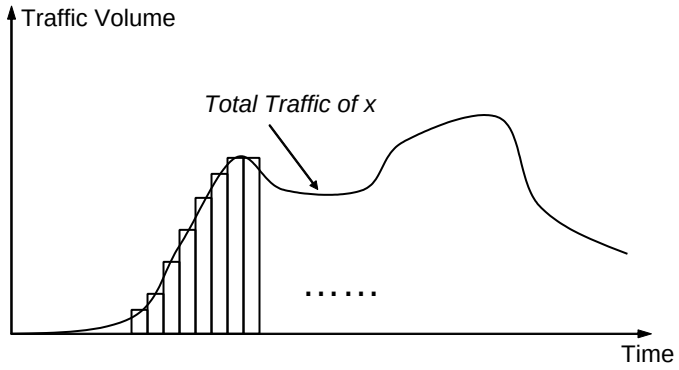


Figure 11: Sampling-based Traffic Analysis

Table 1: Notations used in Sec. III

$\mathcal{V}$	wireless node set
$\mathcal{E}$	edge set
g	gateway node
x	destination node
y	observing node
X	random variable describing x's traffic pattern
Y^X	random variable describing x's traffic pattern observed by y
$H(X)$	entropy of X
$H(Y^X)$	entropy of Y^X
$I(Y^X, X)$	mutual information between X and Y^X

Basic Definition

Ideally, we view the traffic of x as a continuous function of time, as shown in Fig. 11. In practice, the traffic analysis is conducted by dividing time into equal-sized sampling periods, then measuring the amount of traffic in each period, usually in terms of number of packets, assuming the packet sizes are all equal. Therefore, as the first step, we discretize the continuous traffic curve into a piece-wise approximation of discrete values, each denoting the number of packets destined to x in a sampling period.

Now, we use X as the random variable of this discrete value. Y^X is the random variable representing the number of packets destined to x observed at node y in a sampling period. We denote $P(X = i)$ as the probability that the random variable X is equal to i ($i \in \mathcal{N}$) (i.e., the probability that node x receives i packets in a sampling period). Likewise, $P(Y^X = j)$ is the probability that Y^X is equal to j ($j \in \mathcal{N}$), i.e., j packets destined to x go through node y in a sampling period.

Then the discrete Shannon entropy of the discrete random variable X is

$$H(X) = -\sum_i P(X = i) \log_2 P(X = i) \tag{4}$$

$H(X)$ is a measurement of the uncertainty about outcome of X. In other words, it measures the information of node x's traffic (*i.e.*, the number of bits required to code the values of X). $H(X)$ takes its maximum value when the value of X is uniformly distributed. On the other hand, if the traffic pattern is CBR (constant bit rate), then $H(X) = 0$ since the number of packets at any sampling period is fixed[2]. Similarly, we have the entropy for Y^X as follows.

$$H(Y^X) = -\sum_j P(Y^X = j) \log_2 P(Y^X = j) \tag{5}$$

[2]This offers the information-theoretic interpretation for traffic padding: by flattening the traffic curve with blank packets, the entropy of observable traffic is reduced to 0, which perfectly hides the information of the original traffic pattern.

Mutual Information

We define the conditional entropy of random variable X with respect to Y^X as

$$H(X|Y^X) = -\sum_j P(Y^X = j) \sum_i p_{ij} \log_2 \ p_{ij} \tag{6}$$

where $p_{ij} = P(X = i|Y^X = j)$ is the probability that $X = i$ given condition that $Y^X = j$. $H(X|Y^X)$ can be thought of as the uncertainty remaining about X after Y^X is known. The joint entropy of X and Y^X can be shown as

$$H(X, Y^X) = H(Y^X) + H(X|Y^X) \tag{7}$$

Finally, we define the mutual information between X and Y^X as

$$\begin{aligned} I(Y^X, X) &= H(X) + H(Y^X) - H(X, Y^X) \\ &= H(X) - H(X|Y^X) \end{aligned} \tag{8}$$

which represents the information we gain about X from Y^X.

Returning to the example in Fig. 10, let us assume that the observing node y is located on one route destined to x. Since the traffic shape observed at y is the same as x, at any sampling period, if $Y^X = j$, then X must be equal to a fixed value i, making $P(X = i|Y^X = j) = 1$. According to Eq. (6), this makes the conditional entropy $H(X|Y^X) = 0$. According to Eq. (8), we have $I(Y^X, X) = H(X)$, implying that from Y^X, we gain the complete information about X.

On the contrary, if Y^X is independent from X, then the conditional probability $P(X = i|Y^X = j) = P(X = i)$, which maximizes the conditional entropy $H(X|Y^X)$ to $H(X)$. According to Eq. (8), we have $I(Y^X, X) = 0$,[3] (i.e., we gain no information about X from Y^X).

[3]By the definition of mutual information, $I(Y^X, X) \geq 0$, with equality if and only if X and Y are independent.

In reality, since Y^X records the number of a subset of packets destined to node x, it can not be totally independent from the random variable X. Therefore, the mutual information should be valued between the two extremes discussed above (*i.e.*, $0 < I(Y^X, X) < H(X)$). This means that node y can still obtain partial information of X's traffic pattern. However, a good routing solution should minimize such mutual information as much as possible for any potential observing node. More formally, we should minimize

$$\max_{Y \in \mathcal{V}-X} I(Y^X, X) \tag{9}$$

the maximum mutual information that any node can obtain about X.

Penalty-based Routing Algorithm

In this section, we propose a penalty-based routing algorithm to achieve our goal of hiding traffic patterns by exploiting the richness of available paths between two nodes in WMN. Specifically, we choose to adopt the *source routing* scheme. Such a choice is enabled by the fact that one node can easily acquire the topology of the WMN it belongs to, which is mid-sized (within 100 nodes) and static.

When designing the algorithm, we also keep in mind the need to compromise between sufficient security assurance and acceptable system overhead. We show in our algorithm that system performance is satisfactory and security assurance is adequate. Shown in Tab. 2, the algorithm operates in three phases, *path pool generation*, *candidate path selection* and *individual packet routing*. The notations used in Sec. III are listed in Tab. 3.

First, in the path pool generation phase, we generate a large set of diversified routing paths connecting the gateway g and the destination node x, denoted as S_{paths}. The path generation algorithm is an iterative process of applying PBSP (Penalty-Based Shortest Path), a modified version of Dijkstra's algorithm. The PBSP algorithm is shown in the first part of Tab. 2. Here, each node is assigned a penalty weight, and the weight of an edge is defined as the weighted average of penalty weights of its two end nodes. The weight (or cost) of a path is defined as the sum of penalty weights

of all edges constituting this path. The algorithm runs in iterations. Initially, we set the penalty weight of each node as 1, then run the Dijkstra's algorithm to find the first shortest path from the gateway g to x. Next, we increase the penalty weight for each node on this found path. This will make these appeared nodes less competitive to other nodes in becoming components of the next path. After this, the algorithm proceeds to the next iteration, generating the second path, and all nodes appearing on the second path are penalized through increasing their weights. This process iterates until a sufficient number of paths are found. Second, in the candidate path selection phase, we try to choose a combination of diversified routing paths, a subset of paths from the set S_{paths} , denoted as $S_{selected}$. The paths in $S_{selected}$ are selected randomly from S_{paths}. After each choice of a path is placed into $S_{selected}$, the probability factor of that path is decreased to lower the chance of multiple identical paths existing in $S_{selected}$. $S_{selected}$ is changed and renewed corresponding to network activities. Third, in the packet routing phase, we choose randomly from $S_{selected}$ one path for each packet and increase the counter for the selected path subset $S_{selected}$. This $S_{selected}$ path subset expires after a counter reaches its predetermined threshold. Then $S_{selected}$ is renewed by calling the second phase again.

Since packets are assigned a randomly chosen path, and all these candidate paths are designed to be disjoint, the chance that packets are routed in similar paths is small. Our experimental results confirm this intuition. This algorithm is designed to balance the needs of routing performance (finding paths with smallest hop count) and preserving traffic pattern privacy (finding disjoint paths). The penalty weight update function serves as the tuning knob to maneuver the algorithm between these two contradictory goals. During the initialization, when the penalties of all nodes are equal, the path found by the algorithm is indeed the shortest in terms of hop count. As a node is chosen by more routes, its penalty weight monotonically increases, making it less likely to be chosen again. Thus, as the algorithm proceeds, the newly-chosen paths (shortest in terms of its aggregate penalty weight) become more disjoint from existing paths, but longer in terms of hop count. The pace of such shift from "smallest hop-count path" to "disjoint path" is controlled by

how fast the penalty weight update function grows. Our experimental results confirm this reasoning. Finally, by randomly assigning packets along different paths, the algorithm maximally disturbs the traffic pattern of any $g - x$ pair.

Although penalty-based routing has been used in existing literature [12], we are using it for different objects. Their links were penalized for losses or malicious behavior while our approach applies it to avoid using links repeatedly to get better path diversity.

Experimental Results

Simulation Setup

We base our simulations on a randomly generated topology (Fig. 12) (600 x 600) with 30 nodes. The effective distance between two nodes is set to be 250. The whole process of simulation consists of 400,000 logical ticks. In each single tick, a packet is generated at gateway node 0 and its destination is randomly decided to be one of the other 29 nodes. To better simulate real network traffic, we set the probability of 0.05 that at one tick no packet is generated (i.e., idle probability). The distance delay factor is chosen to be 0.003 tick and the hop delay factor is decided as 0.05 tick. We approximate hop delay at any node by multiplying the hop delay factor with its usage count by all paths chosen initially.

With a relatively small node set, we choose 50 as our $PathPoolSize$ and 5 as $SelPathNum$. The selected path subset $S_{selected}$ for any destination node is renewed after sending 50 packets to that node. To obtain multiple diversified paths with Dijkstra's algorithm more quickly, we introduce an exponential penalty function on tag of one node and used γ as the parameter of an exponential function when deciding on which edge to include in a candidate path. To slow down the growing rate of exponential penalty function, we multiply the exponential function with a factor α when calculating $EdgePenalty$. To avoid getting too many identically paths in the beginning stages, we amplify the influence of another node by multiplying tag of another node with β. The penalty parameters α, β, γ are chosen to be 0.5, 15, and 1.85, respectively.

Table 2: Penalty-based Routing Algorithm

/*Penalty-Based Shortest Path*/
$PBSP(Snode, Dnode)$
  For each node $v \in \mathcal{V}$
    $d[v] \leftarrow \infty$
    $prev[v] \leftarrow \infty$
    $visited[v] \leftarrow 0$
  $d[SNode] \leftarrow 0$
  Repeat
    Get unvisited vertex v with the least $d[v]$
    If $d[v] \geq \infty$, **Then** v unreachable
    Else $visited[v] \leftarrow 1$
    For all v's neighbors w
      $EdgePenalty = \alpha[pow(\gamma, (w.tag))] + \beta(v.tag)$
      If $d[w] > d[v] + EdgePenalty$
        $d[w] \leftarrow d[v] + EdgePenalty$
        $prev[w] \leftarrow v$
  Until $visited[v] = 1, \forall v \in \mathcal{V}$

/*Generate S_{paths} for each $g - x$ pair*/
$GenPath()$
  For all non-gateway nodes x
    For each node $v \in \mathcal{V}$
      $v.tag \leftarrow 1$
    Repeat
      PBSP(g, x)
      Get new $g - x$ path P_{new} from vector $prev[]$
      Store P_{new} in S_{paths}
      For all nodes v on P_{new}
        $v.tag \leftarrow v.tag + 1$
    Until *PathPoolSize* paths found.

/*Select $S_{selected}$ for each $g - x$ pair*/
$SelPath()$
  Repeat
    $rnd = rand()\ mod\ PathPoolSize$
    select rndth path from S_{paths}
  Until $SelPathNum$ paths selected

/*Decide path for arriving packet*/
$RoutePkt(Snode, Dnode)$
  $Packets[Dnode] \leftarrow Packets[Dnode] + 1$
  $rndpath = rand()\ mod\ SelPathNum$
  route packet along the $rndpath$th path from $S_{selected}$
  If $Packets[Dnode] > ReSelPathCnt$
    $Packets[Dnode] \leftarrow 0$
    SelPath()

Table 3: Notations used in Sec. III

v, w	node
$v.tag$	number of times v is included by a path
α	factor to slow down penalty rate
β	factor to avoid many identical paths in the beginning stages of path generation
γ	base of exponential penalty function
$d[]$	penalty vector for every node
$prev[]$	vector to store P_{new} in reverse order
$Packets[]$	vector to store the number of arrived packets for every node

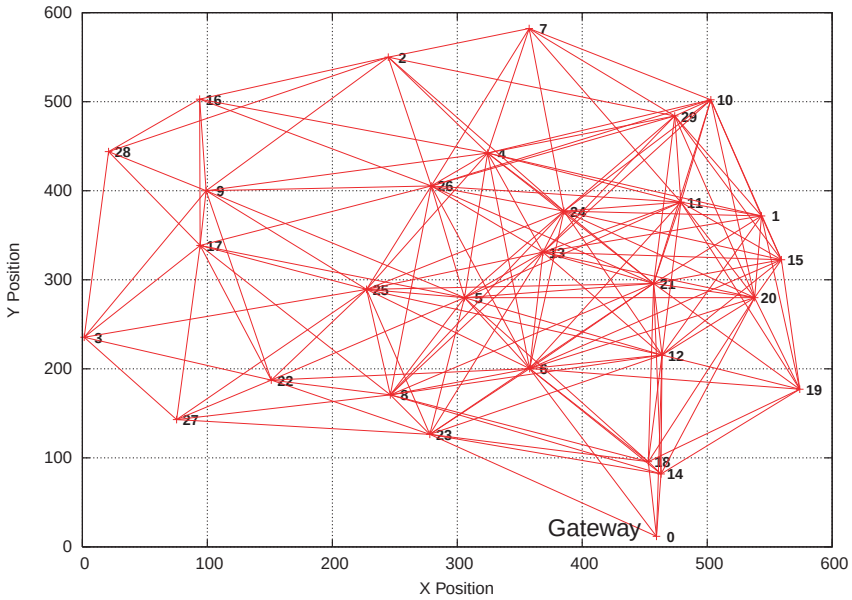
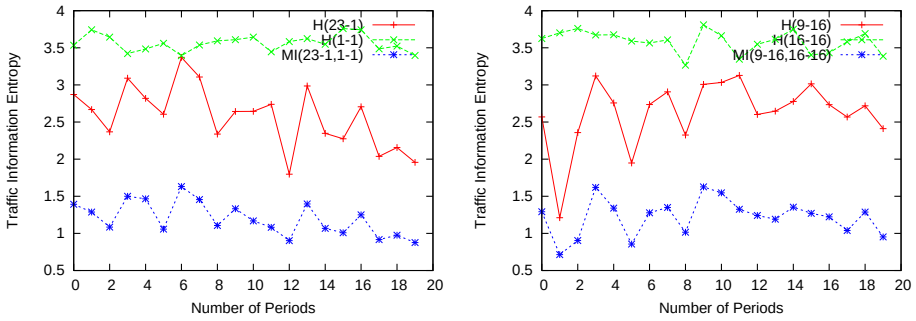


Figure 12: Experimental Topology

Traffic Entropy and Mutual Information

The total 400,000 ticks are divided into 20 periods. Each period is then divided into 50 intervals and one interval is 400 ticks long. Within each interval, for each destination node x, we count the number of packets that all other nodes y have relayed for x. Then for each period, we independently calculate the traffic entropies $H(X)$, $H(Y^X)$, and mutual information $I(Y^X, X)$ based on their definitions in Sec. III.

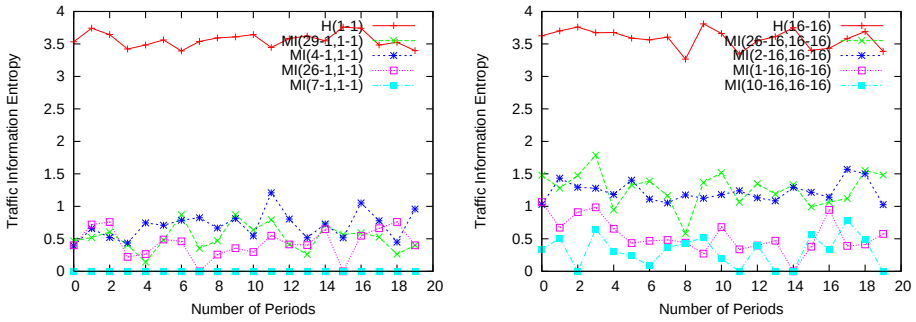


(a) Destination: Node 1, Observer: Node 23 (b) Destination: Node 16, Observer: Node 9

Figure 13: Traffic Entropy along Time (Single Observer, $\gamma = 1.85$)

Due to the space limit, we only show part of our results. Among all nodes in the network, we choose two sets of nodes. Nodes in the first set $\{1, 6, 11, 15, 23, 24, 25, 29\}$ are close to (2 to 3 hops) the gateway node 0. Nodes in the second set $\{2, 3, 7, 16, 17, 28\}$ are at the edge of the network, 4 to 5 hops away from the gateway. We choose two representative nodes, 1 and 16, out of each set.

Fig. 13 shows the variance of traffic entropy and mutual information as a function of the time. In Fig. 13 (a), $H(1-1)$ denotes the traffic entropy of node 1. $H(23-1)$ denotes the traffic entropy of node 23 based on its observation on node 1. $MI(23-1, 1-1)$ denotes the mutual information that node 23 shares with node 1. The same notation rules apply for Fig. 13 (b), where node 16 is the destination, and 9 is the observer. In both pictures, the observing node only shares 40% or less of the total information about the observed destination node at any sampling period.



(a) Destination: 1, Observers: Node 4, 7, 26, 29 (b) Destination: 16, Observers: Node 1, 2, 10, 26

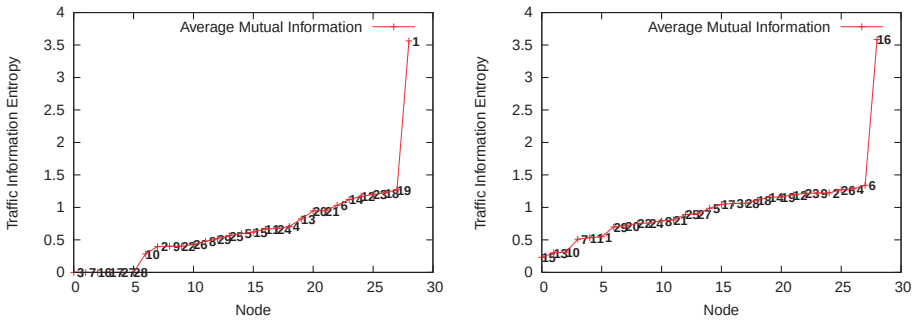
Figure 14: Traffic Entropy in Different Sampling Periods (Multiple Observers, $\gamma = 1.85$)

This observation is further confirmed in Fig. 14, where we plot the time-variant mutual information that destinations 1 and 16 share with other randomly-chosen observing nodes. These results show that with our algorithm, the destination node is able to consistently limit the proportion of mutual information it shares with the observing nodes.

Which Nodes have more Mutual Information?

In Fig. 15 (a), we calculate the time-averaged mutual information for all observing nodes with respect to the destination node 1. (The nodes are sorted in ascending order.) Here, we observe an almost linearly-growing curve except at its head and tail. For nodes at the head of the cure, their mutual information is 0 since they lie at the outer rim of the network. Hence, they are not chosen by our routing algorithm to relay traffic for node 1. At the tail of the curve is destination node 1, whose mutual information is actually the traffic entropy of its own. In Fig. 15 (b), we observe the same phenomenon for destination 16, except at the head of the curve. This is because its network location is opposite to the gateway, making every node of the network to be its candidate relaying node.

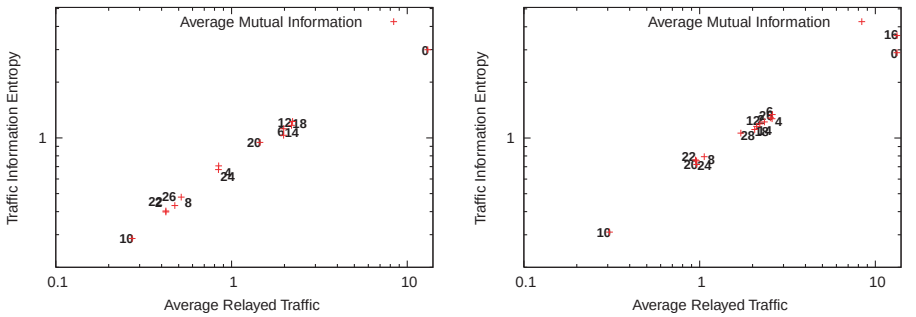
This leads us to investigate whether such distribution of mutual information is related with other factors. We considered the mutual information of each node with certain metrics, such as its distance to the destination. However, we failed to find any causal relationship. We also considered sorting the observation nodes based on their averaged relayed traffic (i.e., the average number



(a) Destination: Node 1 ($\gamma = 1.85$) (b) Destination: Node 16 ($\gamma = 1.85$)

Figure 15: Sorted Traffic Mutual Information

of packets that each node relays in a sampling period) on a log-log scale. We found a linear distribution as shown in Fig. 16.



(a) Destination: Node 1 ($\gamma = 1.85$) (b) Destination: Node 16 ($\gamma = 1.85$)

Figure 16: Power-law Correlation of Mutual Information and Amount of Traffic Relayed

Obviously, such a power-law correlation tells us that the more traffic an observing node relays for a destination node, the more mutual information can be obtained about its traffic entropy. Furthermore, it gives us one way to experimentally quantify the relationship of these two metrics. Let T be the amount of traffic relayed and I be the mutual information, then their power-law relationship can be written as

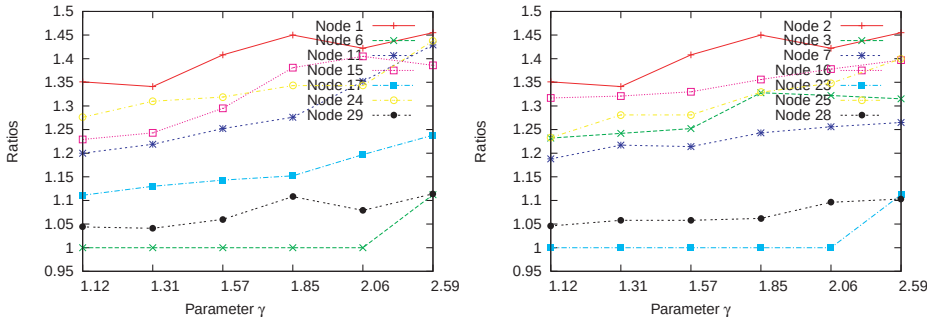
$$I = aT^k \tag{10}$$

where a is the constant of proportionality and k is the exponent of the power law, both of which can be measured from Fig. 16. If $k < 1$, then the mutual information of an observing node grows in a sub-linear fashion to the amount of its relayed traffic increases. If $k \geq 1$, this mutual information

grows in a super-linear fashion. From Fig. 16 and the same results for other destination nodes, we have $k < 1$. This implies that an observing node has to relay more and more traffic each time, in order to make its mutual information further grow with the same increment.

Trade-off between Performance Degradation and Traffic Privacy

Finally, we study the performance trade-off of our algorithm by tuning its exponential penalty function base γ. The performance degradation introduced by our algorithm is captured by the average hop ratio. For each gateway-destination pair $g - x$, this metric is defined as the ratio between the average number of hops a packet goes through using our algorithm and the number of hops of the shortest path between g and s. From Fig. 17, we can see that the average hop ratio increases as γ increases. The direct neighbors of the gateway are less sensitive to the change of γ. (See node 6 in Fig. 17(a) and node 23 in Fig. 17(b).)

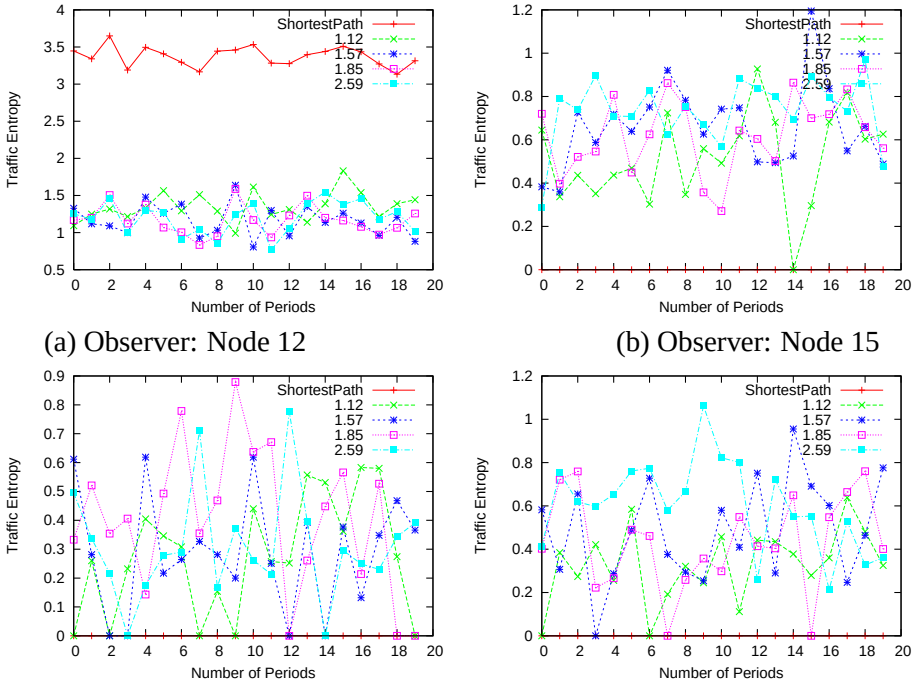


(a) Hop Ratio of Nodes in the First Set (b) Hop Ratio of Nodes in the Second set

Figure 17: Average Hop Ratio

In Fig. 18 and Fig. 19 we find that, under shortest path routing, the mutual information of a node is 0 if it is not on the path to the destination node. Otherwise, the mutual information of a node is much higher than when using the new algorithm. Also worth noting is the observation that increasing γ has different impact on different nodes, depending on its distance to gateway, destination, and its location in the WMN. Consider node 12 (Fig. 18) and 6 (Fig. 19). Since they lie near the gateway node and are relatively centrally situated, their observed mutual information varies little with respect to changes of γ. In contrast, for node 22 (Fig. 18), which is far away

from destination node 1 and on edge of WMN. The mutual information shared between itself and node 1 increases with the growth of γ, indicating more traffic is routed through farther nodes. This tendency of routing packets via farther nodes leads to a higher average number of hops, which is confirmed by our analysis of the average hop ratio.



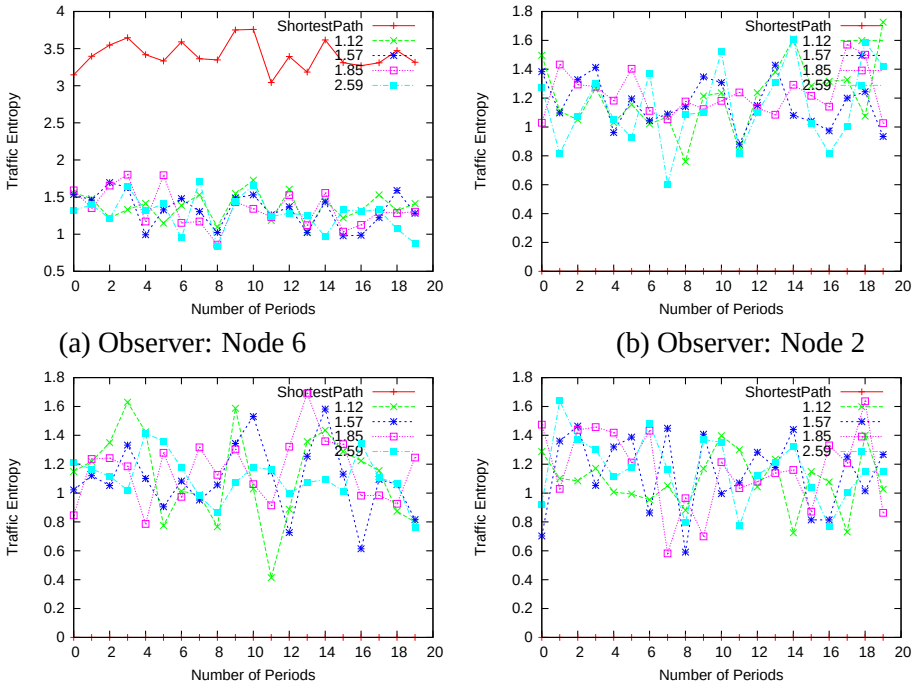
(a) Observer: Node 12 (b) Observer: Node 15

(c) Observer: Node 22 (d) Observer: Node 26

Figure 18: Traffic Mutual Information under Different Penalty Parameters (Destination: Node 1)

However, traffic mutual information tends to decrease once γ gets too high (2.59 in Fig. 18). This is due to the fact that when penalty values of many possible edges get large quickly, their relative differences become less. Consequently, the number of candidate paths decreases. The fluctuation of node 26 (Fig. 18) is due to its position in center of the topology and being equi-distance between the gateway and destination nodes. Similar observations can be made about the mutual information values of destination node 16 (Fig. 19).

We observe from Fig. 20 that the algorithm achieves our goal of preserving traffic patterns. It is easy to conclude that in normal shortest path routing, all relaying nodes share the same traffic information with the destination node, as shown by the tail of the *ShortestPath* curve in Fig. 20.



(a) Observer: Node 6

(b) Observer: Node 2

(c) Observer: Node 14

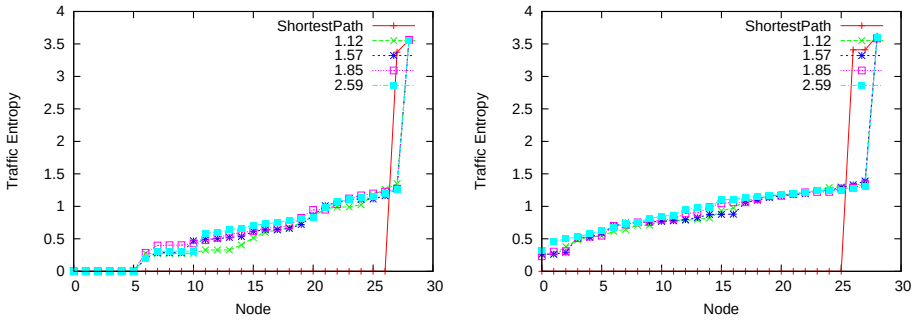
(d) Observer: Node 19

Figure 19: Traffic Mutual Information under Different Penalty Parameters (Destination: Node 16)

However, for our algorithm, the mutual information shared between relaying nodes and the destination node varies less among the relaying nodes. As γ increases, the more leveled off the curve becomes, and the closer we are to the goal of minimizing the greatest mutual information, as formulated in Eq. 9. It is also interesting to observe that the mutual information is 0 for some nodes far away from both the gateway and the destination nodes, e.g., Fig. 20 (a), when destination is 1. In contrast, all nodes participate in relaying packets for destination 16 (Fig. 20 (b)), since destination and gateway nodes are in opposite directions with respect to WMN topology.

Collusion Analysis

The relative small size of a typical WMN makes it easy for spatially close eavesdroppers to find each other. This is concerning since there is a higher possibility of collusion of two malicious observers by exchanging their observed traffic patterns. This motivates us to make our proposed



(a) Destination: Node 1 (b) Destination: Node 16

Figure 20: Sorted Traffic Mutual Information under Different Penalty Parameters

solution resilient to such collusion threats. To analyze the extent to which collusion reveals information about the original traffic pattern, we study the fluctuation of the observed traffic information. In this way, we can analyze how much additional information the colluders can collect about the original traffic.

Problem Description

In the first part of this chapter we focused on traffic confidentiality. We studied the problem of traffic pattern concealment via routing control. However, the relative small size of a WMN, aided by the stationary adjacent routers, invites a higher possibility of collusion between several observing relaying routers in the network. Since it is highly possible that different observers will know about various "ups and downs" of target's traffic, if malicious observers interchange their observed traffic information of target users, the combined observations could reveal significant information about the original traffic pattern. This is illustrated in Fig. 21. Given the size of the network (e.g., less than 100 neighbor nodes), we think it is more common that less than two malicious observers exist simultaneously. Hence we focus on analyzing the collusion problem of two observers in this work.

The parameters that affect significantly our collusion analysis include the choice of cooperating observers and the destination target node. Since any routing algorithm will largely depend on the topology of the network and the relative positions of the observers, the source and destination

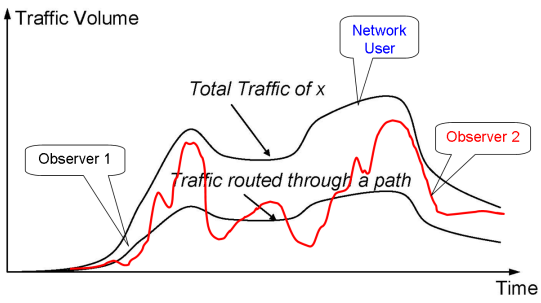


Figure 21: Collusion Reveals Significant Portion of Original Traffic Pattern.

nodes can affect portions of the revealed traffic pattern greatly. Another important parameter is the base of the exponential penalty function explained in Sec. III.

Colluded Traffic Mutual Information

Our modeling of colluded traffic analysis tries to study the influence of collusion to observed traffic patterns of every period. This can help us to evaluate the resilience of our proposed PBSP routing algorithm against collusion attack. In what follows, we consider three nodes x, y, and z. x is the destination node of the traffic from the gateway g to x. Nodes y and z are the observing nodes, which relay packets for x, and also try to analyze the traffic of x. Due to the uncertainty of routing, y and z may or may not be on the same path over time.

Initially, we identify a metric to capture colluded observations. Based on definition of traffic mutual information defined in Sec. III, we measure the colluded observation about destination x with mutual information between x and (y, z). The traffic observations by y and z together can be deemed as a joint distribution of variable Y^X and Z^X. The colluded traffic mutual information $I(Y^X, Z^X; X)$ of random variable (Y^X, Z^X) with respect to X can then be defined as

$$I(Y^X, Z^X; X) = H(Y^X, Z^X) + H(X) - H(Y^X, Z^X, X) \tag{11}$$

where $H(Y^X, Z^X, X)$ is the joint entropy of Y^X, Z^X, and X. $I(Y^X, Z^X; X)$ represents the information gained about X from (Y^X, Z^X) (i.e., from y and z acting together). Their relationship is shown in Fig. 22.

Table 4: Notations used in Sec. III.

$\mathcal{V}$	wireless node set
$\mathcal{E}$	edge set
g	gateway node
x	destination node
y, z	observing nodes
X	random variable describing x's traffic pattern
Y^X, Z^X	random variables describing x's traffic pattern observed by y and z, separately
(Y^X, Z^X)	random variable describing x's traffic pattern observed by y and z together
$H(X)$	entropy of X
$H(Y^X)$	entropy of Y^X
$H(Y^X, Z^X, X)$	joint entropy of Y^X, Z^X, and X
$I(Y^X; X)$	mutual information between X and Y^X
$I(Y^X, Z^X; X)$	colluded mutual information between X and (Y^X, Z^X)

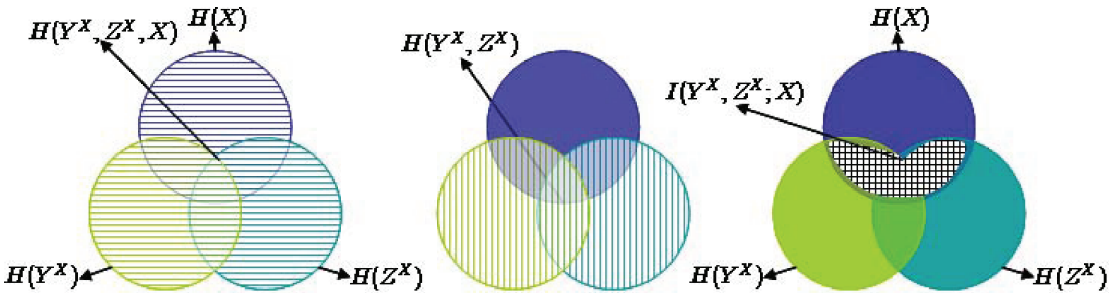


Figure 22: $I(Y^X, Z^X; X), H(Y^X, Z^X)$ and $H(Y^X, Z^X, X)$ in Venn Diagram.

For ease of notation in the following discussion, we use $H(Y, X)$ to denote $H(Y^X, X)$ (i.e., the entropy of traffic that y observes about x). Similarly, we simplify the joint traffic entropy $H(Y^X, Z^X)$ as $H(y, z, x)$, where Y^X, Z^X denote the portions of traffic that Y and Z observe about X. In a subtly different way, we denote $I(Y^X; X)$ as $I(Y; X)$ and $I(Y^X, Z^X; X)$ as $I(Y, Z; X)$.

Traffic Curves

We first present the measured traffic curves as a function of time. In Fig. 23, node 1 is the destination node. We easily conclude that its traffic (i.e., node 1 observing itself) is always the largest. This is because any node can observe the complete traffic of itself while other nodes can only observe a portion of it.

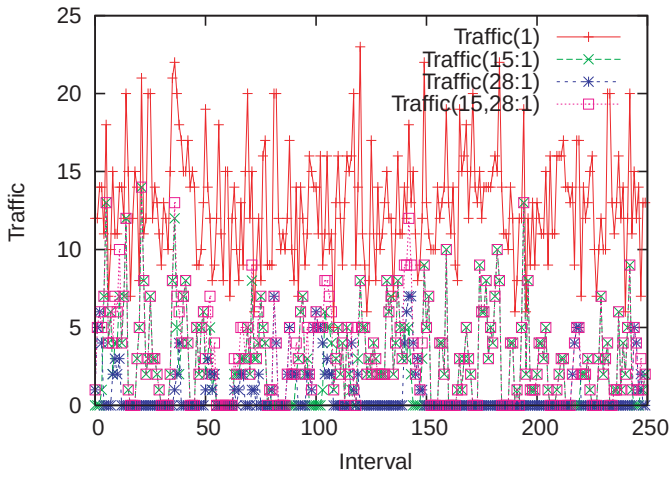


Figure 23: Sampled Traffic Curves from Experiment.

Another observation is the fact that the colluded knowledge about traffic activity of node 1 (in squares), as expected, is higher than any single observer, either node 15 or node 28. Moreover, we confirm that, although, generally speaking, node 15 observes much more traffic of node 1, during some intervals, node 28 out-performs node 15, which increases the aggregated knowledge about node 1's total traffic activity. Example intervals are those near interval 100 and 150.

Colluded Traffic Mutual Information: Single Pair of Observers

The next results are the comparisons of colluded traffic mutual information ($I(y, z; x)$), single observer mutual information ($I(y; x)$ and $I(z; x)$), original traffic entropy ($H(x)$), separately observed traffic entropy ($H(y, x)$ and $H(z, x)$) and joint entropy ($H(y, z, x)$).[4] From our analysis in Sec. III, we conclude the following relationships among these values:

1. $H(y, x), H(z, x) \leq H(y, z, x) \leq H(x)$;
2. $I(y, x), I(z, x) \leq I(y, z, x) \leq H(x)$;
3. $I(y, x) \leq H(y, x) \leq H(x)$;
4. $I(z, x) \leq H(z, x) \leq H(x)$;

We can verify that the simulation results shown in Fig. 24 satisfy these relationships. This means our modeling of traffic activity not only characterizes the traffic pattern fluctuation across the time, but it also actually illustrates the collusion problem. The simulation results of our model conforms to our conjectures.

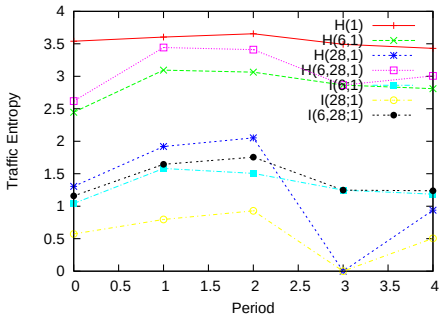
The overlapping curves in Fig. 24(b) indicates that node 23 does not observe any traffic of node 1. This is true since nodes 23 and 1 are on opposite sides of the network. Fig. 25 shows similar results, where node 16 is the destination.

Colluded Traffic Mutual Information: Multiple Pairs of Observers

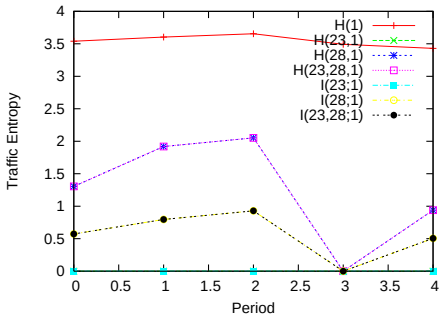
The simulation results confirm the necessary relationships listed previously. We now analyze how collusion affects the performance of Penalty-based Shortest Path (PBSP) routing. To accomplish this, we study the colluded traffic mutual information of several pairs of observers. In this way, we compare the ratio of traffic information collected by different pairs of observers.

From Fig. 26 we observe that the conditions listed above still hold. Additionally, based on the average values of the colluded traffic mutual information curves in both figures, we infer that the

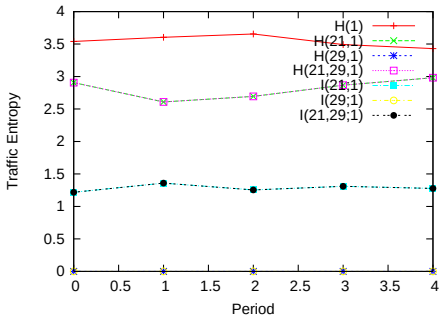
[4]Please note that $H(y, z, x)$, according to our notation, means $H(Y^X, Z^X)$.



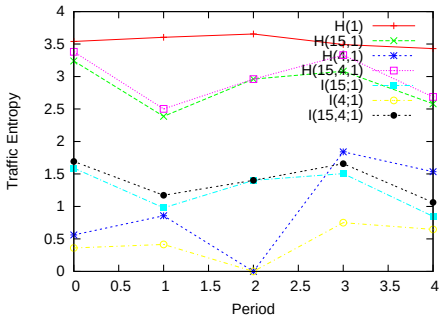
(a) single pair of observers: 6, 28



(b) single pair of observers: 23, 28

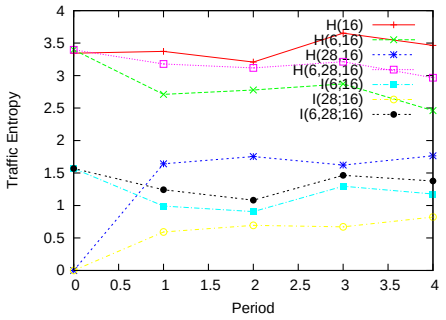


(c) single pair of observers: 21, 29

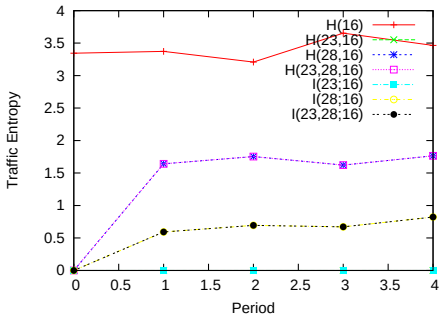


(d) single pair of observers: 15, 4

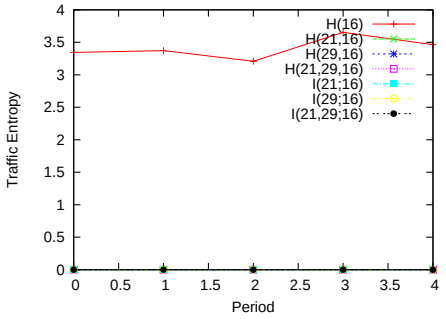
Figure 24: Colluded Traffic Mutual Information (Destination: 1, $\gamma = 1.85$).



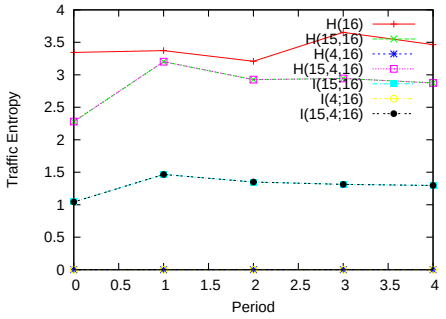
(a) single pair of observers: 6, 28



(b) single pair of observers: 23, 28

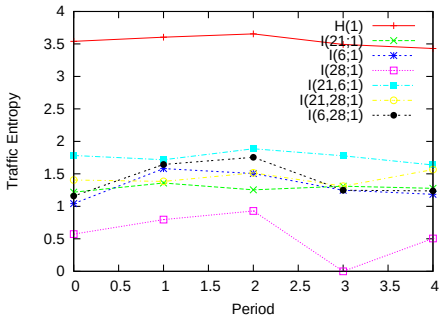


(c) single pair of observers: 21, 29

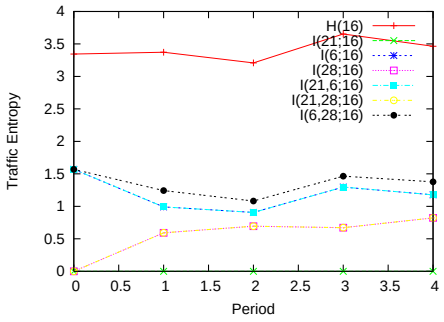


(d) single pair of observers: 15, 4

Figure 25: Colluded Traffic Mutual Information (Destination: 16, $\gamma = 1.85$).



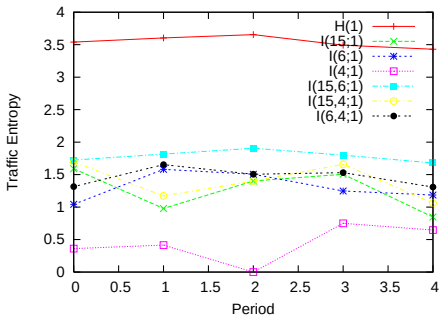
(a) destination: 1, observers: 21, 6, 28



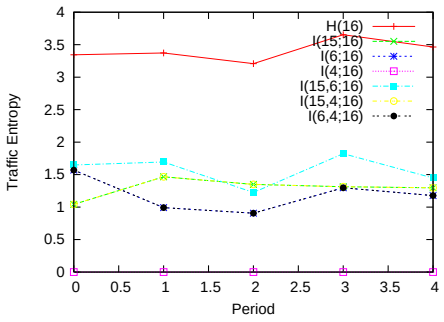
(b) destination: 16, observers: 21, 6, 28

Figure 26: Colluded Traffic Mutual Information (Multiple Pairs of Observers, $\gamma = 1.85$).

PBSP algorithm works well when there are two observers colluding to share their knowledge about one destination.



(a) destination: 1, observers: 15, 6, 4

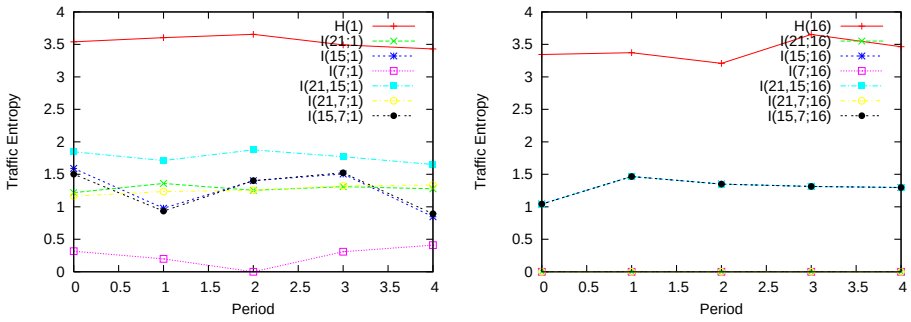


(b) destination: 16, observers: 15, 6, 4

Figure 27: Colluded Traffic Mutual Information (Multiple Pairs of Observers, $\gamma = 1.85$).

To further confirm this conjecture, we examine another set of simulation results, as shown in Fig. 27. The colluded traffic mutual information of all observer pairs in this figure does not exceed half of the total traffic information. In Fig. 27(b), however, we notice some small error in the curves (i.e., the value of $I(15, 6; 16)$ is slightly less than that of $I(15; 16)$ for period 2). Although this is a small error, it is similar to approximation error when computing $H(Y^X, Z^X, X)$. Instead of employing three parallel $PacketCounter$s to get the aggregate traffic information, the simulation program approximates it, based on the packet count value dictionary, which results in a lower $I(Y^X, Z^X; X)$ value.

The same explanation applies to the discrepancy in Fig. 28(a). The average value of colluded traffic mutual information of all observer pairs in Fig. 28 remains approximately less than half of the total traffic entropy of the target node across all time periods.



(a) destination: 1, observers: 21, 15, 7 (b) destination: 16, observers: 21, 15, 7

Figure 28: Colluded Traffic Mutual Information (Multiple Pairs of Observers, $\gamma = 1.85$).

Related Work

Multi-hop wireless mesh networks (WMN) are gaining popularity. Current deployments of WMN either serve as substitutes for traditional WLAN Internet connections, or aim at providing infrastructural large-scale network access [45].

Existing research [6, 32, 14, 11] on WMN has focused on how to better utilize the wireless channel resource and enhance its performance. For example, some researchers derive the optimal node density following a capacity analysis [28], while others devise more efficient [18] protocols. A survey paper by Akyildiz et al. [10] provides a good source for existing and ongoing research about wireless mesh networks. Some of the proposed solutions include equipping mesh routers with multiple radios and distributing the wireless backbone traffic over different wireless channels, routing the traffic through different paths [22, 57], or a joint solution of these two [47, 46]. Theoretical studies show that these approaches can significantly increase the capacity of WMN [36, 34]. These results make significant steps towards enabling WMN as an attractive alternative for broadband Internet access.

Information Theory is widely used and proves to be a useful tool. It applies to situations where variations are frequent and unpredictable. It also helps to identify patterns and the extent of observed variations. Serjantov et al. [51] define an information theoretic anonymity metric and suggest developing more sophisticated probabilistic anonymity metrics. Existing research in the Internet setting employs information theoretical coding [33]. However, such analysis is often too complex and impractical for WMNs. The book by David Mackay provides a good source for background knowledge in information theory [40].

Privacy has been a major concern of Internet users [17, 55]. Two types of techniques have been proposed to preserve user traffic privacy and increase the difficulty for performing harmful traffic analysis [48, 13] in the existing literature of traffic pattern concealment. They are anonymous overlay routing [59, 13, 26, 33, 27, 21, 49] and traffic padding [52]. The former approach provides user anonymity in an end-to-end connection through layered encryption and multi-hop overlay routing. The latter one conceals the traffic shape by generating a continuous random data stream at the link level. However neither approach is applicable to WMN directly. First, the number of nodes in a WMN is limited. Second, the traffic forwarding relationship among nodes is strongly dependent on their locations and the network topology. To better utilize the wireless channel resource and enhance the data delivery performance, a shortest path routing technique is usually selected (or a load-balanced routing scheme is employed). Such observations indicate that the anonymity systems, which rely on relaying traffic among nodes (randomly selected out of thousands) to gain anonymity, can not effectively preserve users' privacy in WMN (or at the cost of significant performance degradation). On the other hand, traffic padding mechanisms consume a considerable amount of network bandwidth, which makes it impractical in resource-constrained WMNs.

The schemes designed for wireless ad-hoc networks [56, 15] are more focused on location and identity privacy. While these are still issues in WMN, the traffic rates and temporal variations are more meaningful and consequential. To the best of our knowledge, no existing work has studied collusion problems about traffic privacy in the scenario of Wireless Mesh Networks.

Conclusion

This chapter identifies the problem of traffic privacy preservation in wireless mesh networks (WMN). To address this problem, we introduced a light-weight architecture for WMN, then proposed "traffic entropy", an information theoretic metric to quantify how well a solution performs at preserving traffic pattern confidentiality. A new penalty-based shortest path routing algorithm was described and analyzed. We evaluated our scheme in the presence of two malicious colluding nodes. Simulation results show that our algorithm is able to preserve traffic privacy, while minimizing the network performance degradation within acceptable ranges. Our simulation analysis demonstrate the resilience of our solution against two colluding observers.

For future work, we will focus on the following problems. First, although our algorithm is evaluated in a single-radio, single-channel WMN setting, it can be easily enhanced to exploit the advantages of multiple radios and multiple channels available in WMNs. Performance evaluation of the enhanced algorithm in such settings should provide interesting results. It is also beneficial to evaluate the possibility of devising a distributed routing algorithm that achieves the same goal but which supports better scalability.

CHAPTER IV

PRIVACY PRESERVATION IN WIRELESS SENSOR NETWORKS

Preserving information confidentiality is a critical issue for wireless sensor networks. While existing security solutions (e.g., encryption) could protect the data content, they can not protect against direction-based traffic analysis. Preserving directional traffic privacy is a challenging problem for wireless sensor networks, as the conventional approaches such as traffic padding and routing control are usually very resource-consuming. This chapter investigates the effectiveness of privacy preserving mechanisms and seeks an optimal solution for preserving privacy in a resource-constrained environment. It presents a novel privacy model that characterizes the application-specific impact of pattern revelation. Via this privacy model, the privacy preservation problem is formulated as an optimization-problem, where optimal routing schemes are derived. Through theoretical analysis and simulation validation, we evaluate the performance of the optimal privacy preservation routing scheme and demonstrate its trade-off in privacy preservation and routing efficiency.

Introduction

Wireless sensor networks are formed by a collection of sensor devices which are capable of sensing, data processing, and communicating via wireless medium [29]. They can be readily deployed in diverse environments to collect and process useful information in an autonomous manner. Thus, they have a wide range of applications in the areas of health care, military, and disaster detection. Sensor networks are envisioned to change the way people interact with the physical environment and to have a significant social impact.

One of the most notable challenges that threaten successful deployment of sensor networks is the protection of information privacy. The challenge comes from two characteristics of wireless sensor networks: (1) open wireless medium prone to eavesdropping;and (2) sensors prone to physical capture, which make it vulnerable to a variety of attacks. For a sensor network that provides

surveillance service, information communicated within the sensor networks involves when and where an event happens, and what is the event. This defines two types of information privacy – the contextual privacy (when and where) and the content-wise privacy (what) [30]. It is obvious that the content-wise privacy is a critical issue for sensor networks. The contextual information, however, also reveals important information with respect to the sensor network operation. In an event-driven sensor network, where messages are only generated and sent upon the detection of certain event, the attackers can easily infer the contextual information (location and time of the event) by observing the traffic patterns.

Content-wise privacy is most often protected via message encryption. In the existing literature, the security mechanisms that support the content-wise privacy has been extensively researched (e.g., link-level security solution [31], key distribution [23]). Only limited work, however, has been done on the contextual privacy issues associated with sensor communication. The work of [30], one of the first works on contextual privacy, has studied protecting location information (so called location privacy). In this work, the authors study a sensor network application scenario of panda hunting. They define location privacy and provide a privacy preservation solution via controlled random routing and flooding. Though the work of [30] provides a convincing solution and makes an important step towards location privacy, two major issues in the area of contextual privacy:

- Lack of a precise definition of contextual privacy which is generalizeable into different application scenarios.
- Lack of an analytical model that balances resource requirements of privacy preservation solutions and their effectiveness. Many of the contextual privacy preservation approaches, such as traffic padding and routing control, are quite resource-consuming. For resource-constrained sensor networks, it is important to carefully examine the resource requirements of these solutions and provide a tuning mechanism can trade-off effectiveness against resource requirements.

To address the above issues, this chapter presents an optimization-based theoretical framework that characterizes the effectiveness of privacy preserving mechanisms. Our definition of privacy is general and can be customized to application-specific scenarios. We focus on the location privacy issue in wireless sensor networks in this work.

The chapter seeks an optimal solution for preserving privacy in a resource-constrained environment. It presents a novel privacy model based on an attacker penalty function, which characterizes the application-specific impact of pattern revelation. Via this privacy model, the privacy preservation problem is formulated as an optimization problem where the optimal routing schemes are derived. Through theoretical analysis and simulation validation, we show several important properties of the optimal privacy preservation.

The remainder of this chapter is organized as follows. Sec. IV presents the attacker and privacy models. These models are used further in Sec. IV to formulate privacy preservation as an optimization problem. Sec. IV.[TODO] The simulation results are provided in Sec. IV demonstrates how an optimal routing protocol can be designed to have good trade-off between privacy preservation and network performance. A review of the existing literature is provided in Sec. IV. We present conclusions in Sec. IV.

Model

Sensor Network Model

We consider an event-driven sensor network with a set of sensor nodes $n \in N$ as shown in Fig. 29. The event set $\mathcal{E} = \{e\}$ denotes the set of all possible events in this network. In this chapter, we focus on the events that are characterized by their geographical locations. For example, in the dirty bomb detection and localization scenario [3], the static sensors are deployed around a stadium to report position data about the moving node. Let E be the random variable that represents the detected event in the sensor network. Then the probability that this event is e is denoted as $Pr(E = e)$, or in abbreviation $Pr(e)$.

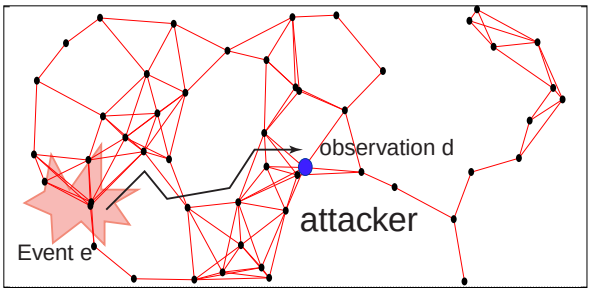


Figure 29: Example Sensor Network.

When an event e is detected in the sensor network, the sensors that detect the event will send messages among themselves and/or to the data sink. We assume that the confidentiality and integrity of the message content are protected via data encryption and message authentication code. If source routing is used in the network, then the source route information carried in the packet header is also protected.

Attacker Model

The attacker tries to infer the event occurred in the sensor network based on his observations. Since the packets are encrypted, the attackers could only do so by observing the traffic patterns (*e.g.*, from which direction packets are coming). Depending on the observation location(s) of the attacker and his observation range, different observations may be made for a single event. Without loss of generality, we assume that an attacker has a fixed set of observation locations and ranges when observing a single event. Thus, for an event e, there is only one possible observation d for a particular attacker. Let us denote the set of all possible observations from attacker A upon different events as $\mathcal{D}_A$. We further denote the set of all possible observations from different attackers with different locations and observation ranges as $\mathcal{D} = \bigcup_A \mathcal{D}_A$.

Upon observing $d \in \mathcal{D}_A$, the attacker A may infer a set of possible events $\tilde{\mathcal{E}} = \{\tilde{e}\}$, and take some corresponding actions. For example, he may deduce that the dirty bomb needs to be moved to a new position. If the real event that occurs is e, then we model the utility of the attacker by inferring event $\tilde{e}$ via an *attacker pay-off function* $S(\tilde{e}, e)$. A positive value of $S(\tilde{e}, e)$ indicates the gain of the attacker when his inferred event is equal to (or close to) the true event; while a negative

value represents the penalty to the attacker if his inference is far away from the true event. Here we give several examples to illustrate the concept of the pay-off function $S(\tilde{e}, e)$.

Example 1.

$$S(\tilde{e}, e) = \begin{cases} 1 & \text{if } \tilde{e} = e; \\ -1 & \text{otherwise.} \end{cases} \tag{12}$$

In this example, the attacker will get a pay-off value of 1 if his inferred event $\tilde{e}$ is the true event e. Otherwise, the pay-off is -1, which reflects the penalty on him to react to the wrong events.

Example 2.

$$S(\tilde{e}, e) = F(|l_{\tilde{e}}, l_e|) \tag{13}$$

where $l_{\tilde{e}}$ and l_e are the locations of events $\tilde{e}$ and e, and $|l_{\tilde{e}}, l_e|$ is the distance between these two locations. In this example, the events are identified by their locations. The attacker's pay-off depends on the distance between the inferred event $\tilde{e}$ and the true event e. The function $F(\cdot)$ could take different shapes which reflect different degrees of sensitivity of revealing the event location to the attacker. See Fig. 30. In this figure, the convex curve represents that an attacker is sensitive to event location revelation.

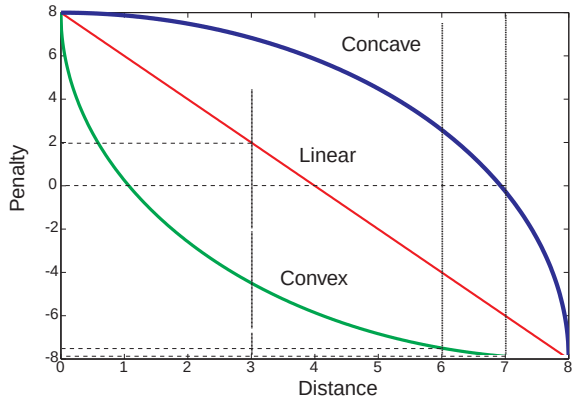


Figure 30: Example Penalty Functions of Different Sensitivity to Event Location Revelation.

When the true event is e, the total pay-off S for the attacker from inferring and reacting to event set $\tilde{\mathcal{E}}$ is

$$S(e, \tilde{\mathcal{E}}) = \sum_{\tilde{e} \in \tilde{\mathcal{E}}} S(\tilde{e}, e) \tag{14}$$

It is obvious that the optimal strategy of a rational attacker is to infer and react to the event set $\tilde{\mathcal{E}}$ so that his total pay-off is maximized, implying:

$$\max_{\tilde{\mathcal{E}}} \sum_{\tilde{e} \in \tilde{\mathcal{E}}} S(\tilde{e}, e) \tag{15}$$

Without knowing the true event e, the attacker determines his strategy by estimating his total pay-off $\hat{S}$ based on his observation d. The estimation is done through the relation between an event and the possible observations by the attacker. Let $Pr(e|d)$ be the probability of event e's occurrence if d is observed. Then, the attacker's pay-off $\hat{S}$ of inference set $\tilde{\mathcal{E}}$ based on observation d is estimated as

$$\hat{S}(d, e, \tilde{\mathcal{E}}) = \sum_{e \in \mathcal{E}} \{Pr(e|d) \times \sum_{\tilde{e} \in \tilde{\mathcal{E}}} S(\tilde{e}, e)\} \tag{16}$$

Thus the attacker's optimal strategy is to derive $\tilde{\mathcal{E}}$ so that $\hat{S}$ is maximized. To do that, he first needs to estimate $Pr(e|d)$. Note that some events in the sensor network may not be observable by attacker A, depending on his observation location and range. We denote the set of events that could be observed by A as $\mathcal{E}_A$. Obviously, $\forall e \notin \mathcal{E}_A, Pr(e|d) = 0$. For an event observable by attacker A, we use $Pr_A(e)$ to denote the probability that this event is e, where $e \in \mathcal{E}_A$. For an observation by attacker A, we use $Pr_A(d)$ to denote the probability that this event is d, where $d \in \mathcal{D}_A$. Based on Bayes' theorem, we have

$$Pr(e|d) = \frac{Pr(d|e) \times Pr_A(e)}{Pr_A(d)} \tag{17}$$

where $Pr(d|e)$ is the probability that event e triggers observation d. We assume that attacker A knows $Pr_A(e)$ for all $e \in \mathcal{E}_A$ as a priori knowledge of the sensor network. The attacker could further derive $Pr_A(d)$ for all $d \in \mathcal{D}_A$ based on his observation. $Pr(d|e)$ depends on the attacker's

observation location and range, as well as how the event-driven messages are routed in the network. We assume that the attacker is able to estimate $Pr(d|e)$ based on his knowledge of message routing mechanisms in the sensor network. We will discuss the details of such estimations in the next section.

The pay-off $J(\tilde{\mathcal{E}})$ for attacker A can be expressed as:

$$J(\tilde{\mathcal{E}}) = \sum_{e \in \mathcal{E}_A} \{ \frac{Pr(d|e) \times Pr_A(e)}{Pr_A(d)} \sum_{\tilde{e} \in \tilde{\mathcal{E}}} S(\tilde{e}, e) \} \tag{18}$$

Then, the attacker's strategy is formulated as

$$\begin{aligned} \mathbb{A}: \quad & \text{maximize} \quad J(\tilde{\mathcal{E}}) \\ & \text{where} \quad \tilde{\mathcal{E}} \subseteq \mathcal{E}_A \end{aligned} \tag{19}$$

Given observation d, the knowledge of $Pr(d|e)$, $Pr_A(e)$ and attacker penalty function $S(\tilde{e}, e)$, the optimal strategy of attacker A is denoted as $\tilde{\mathcal{E}}_A^*(d)$, which leads to the optimal attacker pay-off as $J_A^*(d)$. The property of $\tilde{\mathcal{E}}_A^*(d)$ is given in the following theorem.

Theorem 1, Given observation $d \in \mathcal{D}_A$, the knowledge of $Pr(d|e)$, $Pr_A(e)$, and the attacker penalty function $S(\tilde{e}, e)$, the inferred event set $\tilde{\mathcal{E}}_A^*(d)$ is the optimal strategy for attacker A if and only if $\forall \tilde{e} \in \tilde{\mathcal{E}}_A^*(d)$,

$$\sum_{e \in \mathcal{E}} Pr(d\&e) \times S(\tilde{e}, e) > 0 \tag{20}$$

where $Pr(d\&e)$ is the probability that both d and e occur when an observation is made by attacker A.

Proof: Note that

$$\begin{aligned} & \sum_{e \in \mathcal{E}_A} \{ \frac{Pr(d|e) \times Pr_A(e)}{Pr_A(d)} \sum_{\tilde{e} \in \tilde{\mathcal{E}}} S(\tilde{e}, e) \} \\ = \quad & \frac{1}{Pr_A(d)} \sum_{\tilde{e} \in \tilde{\mathcal{E}}} \sum_{e \in \mathcal{E}_\mathcal{A}} Pr(d|e) \times Pr_A(e) \times S(\tilde{e}, e) \end{aligned} \tag{21}$$

Since $\sum_{e \in \mathcal{E}_A} Pr(d|e) \times Pr_A(e) \times S(\tilde{e}, e)$ are independent for different $\tilde{e}$s, to maximize Eq. (21), an event $\tilde{e} \in \mathcal{E}_A$ should be included in the set $\tilde{\mathcal{E}}_A^*$ if and only if

$$\sum_{e \in \mathcal{E}} Pr(d|e) \times Pr_A(e) \times S(\tilde{e}, e) > 0 \tag{22}$$

which leads to the result. ■

Privacy Model

From the above discussions, the pay-off of an attacker indicates the importance of the information revealed to him. Thus it also reflects the value of the contextual information. The goal of privacy preservation is to minimize the pay-offs of all attackers. Formally, let $\mathcal{A}$ be the set of attackers, where each attacker is identified by his location and observation range. Further let $Pr(A)$ be the probability of attacker A's appearance. We define a *contextual privacy index* P of a sensor network as follows.

$$P = \sum_{A \in \mathcal{A}} Pr(A) \sum_{d \in \mathcal{D}_A} Pr_A(d) \times J_A^*(d) \tag{23}$$

The privacy preservation goal is to minimize P. In order to achieve this goal, the sensor network design controls the distribution of $Pr(d|e), \forall A,$ via different message routing schemes. Formally, the optimal strategy of $\tilde{\mathcal{E}}_A^*$ is a function of the attacker's pay-off function S, and conditional probability vector $Pr(d_A|e_A), e_A \in \mathcal{E}_A$. The optimal aggregated pay-off of attacker A upon observing d_A is J^*. Given the distribution of e_A and d_A, J^* is a function of the attacker's pay-off function S, and the vector $Pr(d_A|e_A), e_A \in \mathcal{E}_A$. Formally,

$$J^*(S, P_{d_A}) = \sum_{e_A \in \mathcal{E}_A} \{ \frac{P_{d_A}(e_A) \times Pr(e_A)}{Pr(d_A)} \sum_{\tilde{e} \in \tilde{\mathcal{E}}^*} S(\tilde{e}, e_A) \} \tag{24}$$

It is the routing protocol designer's goal to minimize $J^*(S, P_{d_A})$ for possible attackers As.

Optimal Location Privacy Preservation

The "sense and aggregate" operation mode makes wireless sensor networks vulnerable to direction based traffic analysis. Fig. 31 illustrates how an attacker can trace an event back to its source location by directional traffic analysis. When an event occurs, represented by a bell in Fig. 31, the message about the event is sent out to the sink of the wireless sensor network. The attacker can then trace back where the message source is by listening to the wireless channels, as shown with reverse arrows. We give a formal description of the privacy preservation problem next.

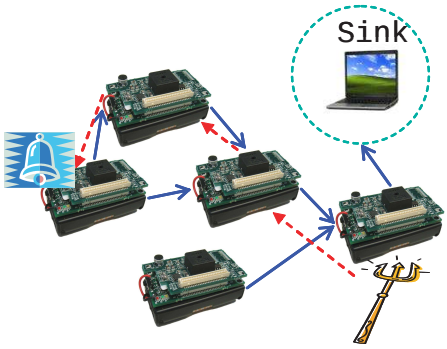


Figure 31: Illustration of Directional Traffic Analysis.

Based on the privacy index defined in Section IV (i.e., Equation 23), we investigate privacy preservation mechanisms that minimize the privacy index P. The design of privacy preservation mechanisms depends on the attackers' knowledge about the routing protocol and network topology. Such information is often publicly available. Thus we will assume that the attacker is able to know the real distribution of $Pr(d|e)$ and $Pr(e)$. Under this assumption, the protection mechanism controls $Pr(d|e)$ (*e.g.*, via routing) to minimize the privacy preservation. This problem is formulated as follows:

$$\min_{Pr(d|e)} \sum_{d \in D} \sum_{\tilde{e} \in \tilde{\mathcal{E}}^*} \sum_{e \in \mathcal{E}} Pr(d|e) \times Pr(e) \times S(\tilde{e}, e, d) \tag{25}$$

Intuitively, the best strategy for routing protocol designer is to maximize the uncertainty about the source event when a particular observation is made about it. We link our formulation with

the concept of information entropy from information theory as follows: when $Pr(d|e)$ follows a uniform distribution, the information entropy is maximized. This leads us to introduce random walk routing to approximate more uniform distributed $Pr(d|e)$.

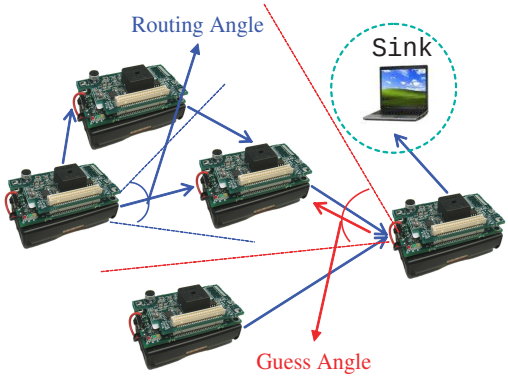


Figure 32: Routing Angle and Guess Angle in Directed Random Walk Routing.

Practically, the routing protocol designer adopts a directed random walk routing to decide the next hop during the routing stage. Directed random walk routing is controlled by the *routing angle (RA)*, which determines the set of sensors from which the next hop sensor is chosen. (The goal of routing is to aggregate the information at the sink.) The attacker, according to our attacker model, infers the possible sources for the related traffic he observes, based on a *guess angle (GA)*. The routing protocol designer seeks to minimize the privacy index by adjusting to the optimal routing angle for a given network and typical traffic profile, given possible guess angles chosen by attackers. Fig. 32 illustrates the routing angle and the guess angle.

Location Privacy Preservation Algorithms and Simulation

In this section, we discuss the algorithms and simulation environment that a routing protocol designer can utilize to best preserve location privacy. We start with directed random walk routing algorithm. A simulation procedure is next described, from which a traffic log is obtained. By analyzing this log, the real distribution of $Pr(e)$ can be calculated when packets are routed under a given *routing angle*. With continued observation about traffic, it is possible for the attackers to know the real distribution of $Pr(d|e)$ and $Pr(e)$.

A directed random walk routing algorithm aims at delivering packets to sensor network sink s by repeatedly selecting neighbors within the range of specified routing angle. For every node i in a sensor network, an angled-neighbor node list AN_i is maintained by selecting nodes x from the original neighbor node list NN_i. These node xs are in the *routing angle* range from i to sink node s. Mathematically, we have $\cos(\text{routing angle}) < \frac{d^2(i,x)+d^2(i,s)-d^2(x,s)}{2d(i,x)d(i,s)}$ and $AN_i \subset NN_i, \forall i$. (Here $d(x, y)$ denotes the distance between node x and node y.) In order to avoid indefinite walking when applying the directed random walk routing algorithm, it is advisable to obtain a set $PathSet_i$ of successful random paths $Path_i^j$ beforehand for any possible event source i. These $Path_i^j$ paths are next used for source routing and are updated when *routing angle* changes. The paths in path sets $PathSet_i$s are regenerated and updated on a regular basis for sensor node i.

We now describe the algorithm to generate a successful directed random walk (*i.e.*, $Path_i^j$), from sensor node i to sensor network sink s. For every hop in the routing process, a next hop node x is selected and compared to the sink s. If x is not the sink node and the total number of hops has not exceeded length of the longest allowed path, the algorithm proceed to generate a new hop x' and test again. This procedure is illustrated in Tab. 5.

Table 5: Directed Random Walk Routing Algorithm

```
/*Get j th random walk path from Snode to Dnode*/
DRWR(Snode, Dnode, j)
    hopCnt ← 0
    curHop ← Snode
    Repeat
        hopCnt ← hopCnt + 1
        If hopCnt > maxCnt
            Return FAIL
        Put curHop to hopCnt th position of Path^j_Snode
        Randomly choose one node x from AN_curHop
        curHop ← x
    Until curHop = Dnode
```

The above discussed algorithm can be applied in the sensor network simulation program. For simplicity, we assume it is equally likely for an event to occur at any sensor nodes in a sensor

network. Whenever an event e occurs at sensor i, a path $Path_i^j$ is selected randomly from path set $PathSet_i$ to route that message. Any sensor nodes y on the path $Path_i^j$ will observe an event occurrence. This observation is denoted as d_y. If y is an attacker, it tries to infer out which sensor is the source of the event. It will first identify the sensor nodes ($Z = \{z\}$) that are in the *guess angle* as illustrated in Fig. 32. For every sensor z identified, a probability p_z can be assigned to indicate the likelihood that z being the source of observation d_y. Depending on the attacker's strategy, the probability distribution of the zs can be either uniform or dependent on their distances ($|l_y, l_z|$) to the observing position. The weighted pay-off S_{d_y} of this single observation d_y for y can then be denoted as: $S_{d_y} = \sum p_z S(z, e)$. The individual pay-off values S_{d_y} are next used to calculate the per-path privacy index PI. In general, if y is a node on path $Path_i^j$, we can represent the probability that y is the attacker when an attacker exists on $Path_i^j$ as p_y. For any path $Path_i^j$, we have its privacy index as $P_{Path_i^j} = \sum p_y S_{d_y}$.

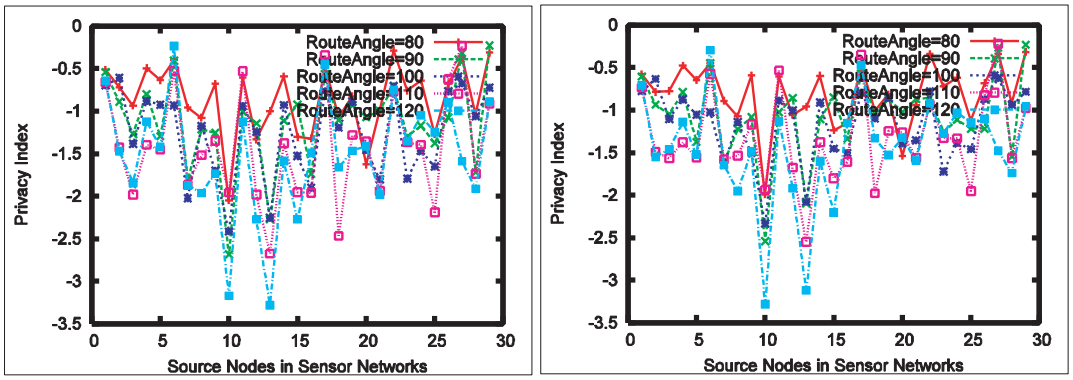
After simulating the routing of messages for sufficient long time, we are able to collect a running log recording the occurrences of all events, as well as the paths used to route the messages to the sink s. By analyzing this log, we can obtain the distribution of probability p_i for an event e to occur at sensor i. With a given event source i, the probability that path $Path_i^j$ is used to route message can also be deduced. We represent it as p_j. By now, we can define the aggregated sensor network privacy index P under *routing angle* and *guess angle* as:

$$P = \sum_i p_i \sum_j p_j P_{Path_i^j} \tag{26}$$

By definition we know that $Pr(e)$ defined in Sec. IV is p_i here. Similarly, $Pr(d|e)$ is an other way of expressing p_j and p_y.

Simulation Results

In this section, we present simulation results of the algorithm discussed in previous section. We simulate the traffic within a wireless sensor network consisting of 30 sensor nodes. Directed random walk routing is used with different routing angles. The obtained logs are further analyzed to calculate the privacy index under various guess angles. We evaluate the effectiveness of privacy preservation and find the optimal routing angle while using directed random walk routing. The pay-off function used is given in Example 1 in Sec. IV.



(a) GuessAngle=30 (b) GuessAngle=60

Figure 33: Privacy Index at Event Source Node.

Fig. 33 plots the privacy index when every individual node serves as the event source. The absolute value of the privacy index when various nodes act as the source depends on the location of the node in the network topology. From Fig. 33 (a) and (b), we infer that the average privacy index tends to increase when the guess angle increases from 30 degrees to 60. This is due to the fact that a larger guess angle leads to more candidate event sources when the attacker is collecting observations. Therefore, it is more likely that the true source location is included in $\tilde{\mathcal{E}}$.

In order to find the optimal routing angle for a given traffic profile (e.g., when every node is equally likely to be the event source), we calculate the average privacy index value for all possible event sources, using different guess angles and various routing angles. The results are illustrated in Fig. 34. We observe that the overall privacy index decreases as the routing angle increases. This

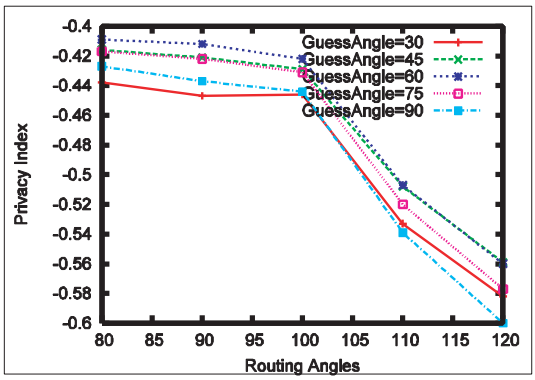


Figure 34: Overall Network Privacy Index under Routing Angle and Guess Angle.

follows from the fact that larger routing angles result in increased randomness for directed random walk routing, but with an associated increase in communication overhead.

From Fig. 34, we notice that the overall privacy index is relatively flat for routing angles less than 100 degrees. We recall that larger routing angles lead to more randomness and to more possible paths. This, in turn, leads to larger average path lengths. By balancing the trade-off between privacy preservation and performance degrade (e.g., throughput, average response time), we infer that , for directed random walk routing to function best, the optimal routing angle is around 100 to 110 degrees in our simulation scenario.

Related Work

Wireless sensor networks have many potential applications in critical scenarios such as battle-field surveillance, environmental monitoring and in-home health care. These missions are sensitive to malicious attacks and demand security protection before large scale deployment of sensors is possible. Security for wireless sensor networks has been studied in the existing literature [54], which includes link layer security [31], broadcast authentication [44], and key management [24]. However, the privacy protection of source location [43, 37] is relatively new research in wireless sensor networks. The paper [20] develops several countermeasures against traffic analysis seeking to locate the source. In [30], the authors formally modeled the source location privacy problem in wireless sensor networks. The routing characteristics of two types of random walk routing protocols are examined. When the source locates in certain regions of the sensing field, the protocols

in [30] suffer performance drop. To address it, [58] proposes self-adjusting directed random walk routing.

Conclusion

This chapter examines the wireless sensor network (WSN) location privacy preservation problem. To address this problem, we quantify the attacker event source guessing pay-offs. Such metrics are accumulated for all possible attackers and guess angles. For various network traffic profiles, we obtain an overall privacy index. This leads to an optimization problem to find the best routing angle, considering the trade-off between privacy and performance (i.e., throughput). We evaluate directed random walk routing schemes under different routing angles by comparing values of our proposed metric via simulation. The result suggests that an optimal routing angle can be found and used in routing protocol design.

For future work, more measurements about privacy and performance trade-off are needed. This includes identifying the inter-relationship of the two with respect to end users and network designers.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

In this thesis, we discuss the privacy demands arising from emerging composite wireless networks. Such privacy demands are multi-faceted and reveal important personal and private information if not properly protected. We classify information privacy to two different types: content-wise privacy and contextual privacy. For content-wise privacy, we adjust and improve the existing tools and solutions for a particular type of privacy protection. For contextual privacy, we propose our own ways to protect it.

By extending existing DRM tools, protection of content-wise data privacy is improved for large scale data distribution. Contextual data privacy is an important issue and is vulnerable to two types of threats: volume-based traffic analysis and direction based traffic analysis. Via simulation experiments, we conclude that while routing control counters volume-based traffic analysis attacks, routing protocol design is needed to protect against direction-based traffic analysis attacks.

As future work, a more general and uniform model of traffic analysis and contextual privacy is needed. We also intend to explore the trade-off between privacy and performance by analyzing it more formally.

BIBLIOGRAPHY

[1] Chaska wireless solutions. http://www.chaska.net/.

[2] Digital invisible ink toolkit (diit). http://diit.sourceforge.net.

[3] Dirty bomb detection and localization. http://www.isis.vanderbilt.edu/Projects/rips/.

[4] Helix drm. http://www.realnetworks.com/products/drm/index.html.

[5] Mesh networks inc. http://www.meshnetworks.com.

[6] Mit roofnet. http://www.pdos.lcs.mit.edu/roofnet/.

[7] Radiant networks. http://www.radiantnetworks.com.

[8] Seattle wireless. http://www.seattlewireless.net.

[9] Winodws media drm. http://www.microsoft.com/windows/windowsmedia/drm/default.mspx.

[10] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Comput. Netw. ISDN Syst.*, 47(4):445–487, 2005.

[11] Mansoor Alicherry, Randeep Bhatia, and Li Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *ACM MOBICOM*, 2005.

[12] B Awerbuch, D Holmer, C Nita-Rotaru, and H Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *ACM Workshop on Wireless Security*, 2002.

[13] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Information Hiding Workshop (IH)*, 2001.

[14] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *ACM MOBICOM*, pages 31–42, 2005.

[15] S Capkun, JP Hubaux, and M Jakobsson. Secure and privacy-preserving communication in hybrid ad hoc networks. Technical Report IC/2004/104, EPFL-DI-ICA, 2004.

[16] Wu chi Feng, Brian Code, Ed Kaiser, Wu chang Feng, and Mickael Le Baillif. Panoptes: Scalable low-power video sensor networking technologies. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 1(2):151–167, 2005.

[17] Roger Clarke. Internet privacy concerns confirm the case for intervention. *Communications of the ACM*, 42(2):60–67, 1999.

[18] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, New York, NY, USA, 2003. ACM Press.

[19] I. J. Cox and M. Miller. A review of watermarking and the importance of perceptual modeling. In *Proceedings of the IS&T/SPIE Conference on Human Vision & Electronic Imaging II*, volume 3016, pages 92–99, San Jose, CA, February 1997.

[20] Jing Deng, Richard Han, and Shivakant Mishra. Countermeasures against traffic analysis attacks in wireless sensor networks. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 113–126, Washington, DC, USA, 2005. IEEE Computer Society.

[21] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium*, 2004.

[22] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM MOBICOM*, pages 114–128. ACM Press, 2004.

[23] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A pairwise key predistribution scheme for wireless sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 42–51, New York, NY, USA, 2003. ACM Press.

[24] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, New York, NY, USA, 2002. ACM Press.

[25] Michael Fahrmair, Wassiou Sitou, and Bernd Spanfelner. Security and privacy rights management for mobile and ubiquitous computing. In *IEEE UbiComp*, 2005.

[26] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *ACM Conference on Computer and Communications Security (CCS)*, 2002.

[27] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2):39–41, 1999.

[28] P. Gupta and P. R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, 2000.

[29] Y. Sankarasubramaniam I. F. Akyildiz, W. Su and E. Cyirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[30] Pandurang Kamat, Yanyong Zhang, Wade Trappe, and Celal Ozturk. Enhancing source-location privacy in sensor network routing. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 599–608, Washington, DC, USA, 2005. IEEE Computer Society.

[31] Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162–175, New York, NY, USA, 2004. ACM Press.

[32] R. Karrer, A. Sabharwal, and E. Knightly. Enabling large-scale wireless broadband: The case for taps. In *HotNets*, 2003.

[33] Sachin Katti, , Dina Katabi, and Katarzyna Puchala. Slicing the onion: Anonymous routing without pki. Technical report, MIT CSAIL Technical Report 1000, 2005.

[34] Murali Kodialam and Thyaga Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *ACM MOBICOM*, 2005.

[35] Purushottam Kulkarni, Deepak Ganesan, Prashant Shenoy, and Qifeng Lu. Senseye: a multi-tier camera sensor network. In *ACM MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, 2005.

[36] Pradeep Kyasanur and Nitin H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *ACM MOBICOM*, pages 43–57, New York, NY, USA, 2005.

[37] Loukas Lazos and Radha Poovendran. Serloc: Robust localization for wireless sensor networks. *ACM Trans. Sen. Netw.*, 1(1):73–100, 2005.

[38] E. Lin, A. Eskicioglu, R. Lagendijk, and E. Delp. Advances in digital video content protection. *Proceedings of IEEE*, 93(1):171–183, 2005.

[39] L.Jiao, Y. Wu, G. Wu, E. Y. Chang, and Y. Wang. The anatomy of a multi-camera security surveillance system. *ACM Multimedia System Journal*, pages 144–163, October 2004.

[40] David J. C. Mackay. *Information theory, inference, and learning algorithms*. Cambridge, Cambridge, 2003 (ISBN: 0-387-95230-6).

[41] G. Miklau and D. Suciu. Controlling access to published data using cryptography. In *IEEE VLDB*, 2003.

[42] M.S.Swanson, M. Kobayashi, and A.H. Tewfik. Multimedia embedding and watermarking technologies. In *Proceedings of IEEE*, volume 86(6), pages 1064–1088, June 1998.

[43] Celal Ozturk, Yanyong Zhang, and Wade Trappe. Source-location privacy in energy-constrained sensor network routing. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 88–93, New York, NY, USA, 2004. ACM Press.

[44] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. Spins: security protocols for sensor networks. *Wirel. Netw.*, 8(5):521–534, 2002.

[45] Krishna Ramachandran, Milind M. Buddhikot, Scott Miller, Kevin Almeroth, and Elizabeth Belding-Royer. On the design and implementation of infrastructure mesh networks. In *Proc. of IEEE WiMesh*, 2005.

[46] A. Raniwala and T. Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *Proc. of IEEE INFOCOM*, 2005.

[47] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *Mobile Computing and Communications Review*, 8(2):50–65, 2004.

[48] Jean-François Raymond. Traffic analysis: Protocols, attacks, design issues and open problems. In *International Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[49] Michael G. Reed, Paul F. Syverson, and David Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1998.

[50] Ravinderpal S. Sandhu. Cryptographic implementation of a tree hierarchy for access control. *Inf. Process. Lett.*, 27(2):95–98, 1988.

[51] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *ACM MOBICOM*, 2002.

[52] W. Stallings. *Cryptography and Network Security*. Prentice Hall, 2003.

[53] Mark Stamp. Risks of digital rights management. *Commun. ACM*, 45(9), 2002.

[54] John P. Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary. *Wireless sensor network security: A Survey*.

[55] Huaiqing Wang, Matthew K. O. Lee, and Chen Wang. Consumer privacy concerns about internet marketing. *Communications of the ACM*, 41(3):63–70, 1998.

[56] Xiaoxin Wu and Bharat Bhargava. Ao2p: Ad hoc on-demand position-based private routing protocol. *IEEE Transactions on Mobile Computing*, 4(4):335–348, 2005.

[57] Yuan Yuan, Hao Yang, Starsky H. Y. Wong, Songwu Lu, and William Arbaugh. Romer: Resilient opportunistic mesh routing for wireless mesh networks. In *Proc. of IEEE WiMesh*, 2005.

[58] Liang Zhang. A self-adjusting directed random walk approach for enhancing source-location privacy in sensor network routing. In *IWCMC '06: Proceeding of the 2006 international conference on Communications and mobile computing*, pages 33–38, New York, NY, USA, 2006. ACM Press.

[59] Li Zhuang, Feng Zhou, Ben Y. Zhao, and Antony Rowstron. Cashmere: Resilient anonymous routing. In *Symposium on Networked Systems Design and Implementation (NSDI)*, 2005.