UNIVERSITY OF CALIFORNIA

Los Angeles

Anonymous and Untraceable Communications in Mobile Wireless Networks

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Jiejun Kong

2004

The dissertation of Jiejun Kong is approved.

Songwu Lu

Jack W. Carlyle

Rajit Gadh

Mario Gerla, Committee Chair

University of California, Los Angeles

2004

To my parents who have made this possible

and also to all the people who have helped or encouraged me

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Problems cannot be solved
at the same level of awareness
that created them.

–Albert Einstein

Wireless networks provide critical supports to enable “anywhere” “anytime” communication services for mobile users. However, the nature of mobile wireless networks makes them very vulnerable to malicious attacks and selfish actions. In civilian wireless networks, service provider can easily locate mobile users and analyze their communication patterns if these users explicitly access services with no anonymity protection. In military wireless networks, anonymity support is even more critical and should be considered as an indispensable network security service. Current military technology has already realized precise weapons to destroy distributed entities within several hundred feet at a time, while the area covered by a wireless hop in many wireless networks is within this fatal range. Consequently, if the enemy is able to obtain definite (though imprecise) information about a mobile node’s location, then the specific target is exposed to great danger. Wireless networking can help legitimate nodes to establish an instant communication structure. Unfortunately, it may also allow passive adversaries to trace network routes and nodes at the end of those routes.

In 1996 A.D., a real world event justified the importance of anonymous communi-

cation. Chechnya rebel leader Dzhokhar Dudayev was killed by Russian armed forces via a traceable wireless call. Before that day, the Russians had tried every physical search methods on the Chechnya battle ground, but failed to locate Dudayev. Wireless communication helped Dudayev to control his army, but also led to his death. Similar to this real world case, consider for example a battlefield scenario with ad hoc, multi-hop wireless communications support. Suppose a covert mission is launched, which includes swarms of reconnaissance, surveillance, and attack task forces. The ad hoc network must provide routes between various task forces, and meanwhile hide their identities as well as motion patterns. Wireless communications in ad hoc networks are essential to coordinate the motion of teams in a mission. But, they can be explored by the adversary to trace the team motion patterns and prepare the counterattack. It is clear that providing anonymity, location privacy, and motion privacy supports for our task forces is critical, else the entire mission may be compromised. This poses challenging constraints on secure routing and data forwarding.

Anonymity and untraceability research was initiated in wired fixed networks. People have already proposed many anonymous schemes to protect communications in wired networks. These schemes are useful in fixed networks. But they are inappropriate or even inapplicable to mobile wireless networks due to at least one of the following reasons:

1. **Fixed and known network status**: Nearly all anonymous schemes designed so far assume that the entire network topology is fixed, and many of them also assume the entire topology is known *a priori*. In DC-Net [29], the network topology is a closed ring and routing is not needed. In Crowds [119] and sorting network [117], pairwise communication has uniform cost (i.e., all nodes are one logical hop away). Protocol can randomly select any member to be next forwarder. This is not true in mobile ad hoc network where multi-hop com-

munication is significantly more expensive than local communication. In MIX-Net [28], a data sender can simply avoid the problem of routing by selecting a random path from the known network topology. All subsequent MIX-Net designs [113] [112] [76] [14] inherit this assumption. In mobile wireless networks, any design built on top of this assumption is invalid due to node mobility and other network dynamics (e.g., offline nodes). In PipeNet [33] and Onion Routing [118], virtual circuit based routing is introduced. However, they assume that network nodes do not move, do not go offline (no solution is proposed to address offline nodes), and the network topology is fixed after initialization. These assumptions are nevertheless inapplicable to mobile wireless networks due to the same reasons. In a nutshell, these schemes treat the underlying network as a simple fixed graph with abundant *a priori* topological information. They do not address mobile routing and do not fit in highly dynamic multi-hop wireless networks.

2. **Impractical cryptographic assumptions**: Longstanding communication links in a fixed network can be protected by *a priori* secure channels because neighborhood status is fixed and predefined. In Onion Routing [118], such secure channels are pre-established in network setup phase rather than anonymous connection setup phase. This assumption is good for fixed networks, but impractical in mobile networks. A major contribution of this work is to devise a reactive (on-demand) method to establish anonymous connections in a highly dynamic mobile network. Dedicated control flows are needed to answer the challenge.

3. **No protection on network topology knowledge**: In fixed networks, routing does not affect network topology which is physically determined *a priori*. However, this is not true in mobile networks where network topology constantly changes due to mobility. A direct porting of anonymity proposals from fixed

networks (e.g., MIX-Net) requires a mechanism to accumulate such network topology knowledge on every sender, thus compromise the anonymity protection of the entire network when a single sender is compromised. Anonymity proposals for mobile networks should not reveal critical network topology information to potential adversarial nodes.

4. **No location privacy and motion pattern privacy support**: In fixed networks, providing location privacy and motion pattern privacy support may be unnecessary since nodes never move. In all existing anonymous schemes, legitimate network members know their (physical or logical) neighborhood. Along a packet flow, each forwarder knows their immediate downstream node and/or immediate upstream node. At least for internal adversary, these schemes fail to ensure location privacy and motion pattern privacy in mobile wireless networks. When location privacy or motion pattern privacy is compromised, mobile nodes are traceable.

5. **Excessive processing overhead**: Many existing anonymous schemes extensively utilize public key cryptography without considering its computational expense on network nodes. This is because the term "efficiency" means polynomially bounded complexity in cryptography, but not its real performance measured on real devices. Despite this is a standard approach in cryptography research, it is not apposite in network security research. In mobile networks, routing is time critical and runs among resource limited mobile nodes. We will use our implementation measurement and simulation study to show that excessive public key crypto-operations significantly degrade routing performance. It is inappropriate to build practical anonymous routing schemes without considering their performance in the real world.

The purpose of this dissertation is to provide anonymity support to mobile wireless

networks in which no existing scheme fits. For mobile wireless networks, we define the concept "mobile anonymity" as our design goal. We show that current anonymous communication schemes are not applicable to mobile wireless networks, and current routing schemes fail to provide anonymity protection. We study the adversary model, the characteristics of attacks against routing schemes in mobile wireless networks, and countermeasures to address the new challenges under the same contexts.

Adversary model is a critical component in any security design, as a solution designed under one adversary model may collapse under another one. To avoid this result we try to build our solution on a comprehensive adversary model. We classify security attacks into two by two categories: passive attacks versus active attacks, and external attacks versus internal attacks. The correspondence between adversary model and the attacks that they can launch is shown in the table below:

	passive attack	active attack
external attack	passive external adversary	active external adversary
internal attack	passive internal adversary	active internal adversary

The line drawn between passive and active attacks is due to the adversary's behavior. The goal of active attacks is to disrupt routing or to stop service provisioning. On the other hand, in the passive case the enemy will avoid aggressive schemes, in the attempt to be as "invisible" as possible, until it traces, locates, and then physically destroys our assets. The line drawn between external and internal attack is due to network membership. External attacks are against wireless links and do not compromise network nodes. Internal attacks are launched by compromised network members. They can utilize internal routing states and cryptographic secrets available on the compromised nodes.

Along the internal-external dimension, a significant contribution of our work is to consider countermeasures against both external and internal attacks. The latter one

is largely unaddressed in previous wireless network security research. In mobile networks, nodes are autonomous units that are capable of roaming independently. This means that mobile nodes with inadequate physical protection are receptive to being captured, compromised, and hijacked. Much easier solutions can be devised if internal attacks are considered infeasible. We will not adopt this elusive approach. In Chapter 3 we present a study on the impact of internal attacks, and the major task of Chapter 5, 6, and 7 is to devise countermeasures against internal adversary.

Along the passive-active dimension, we develop our solutions using a progressive approach. To defend passive attacks, in Chapter 6 we propose an ideal model, *Time Interval and Multi-hop Broadcast Anonymity (TIMBA)*, and prove that TIMBA ensures perfect mobile anonymity against passive external adversary. Then we design and simulate *ANonymous On Demand Routing (ANODR)*, a practical multi-hop on demand routing scheme, as a balance between the ideal TIMBA model and the real world. There is a significant distinction between TIMBA/ANODR and other schemes, such as anonymous schemes like MIX-Net and any existing routing scheme: TIMBA/ANODR is a unique scheme where *no node identity is used in data forwarding and routing*. This ensures perfect identity anonymity in mobile networks, even against internal adversary and active adversary. ANODR is comprised of three variants, each of them trades off security guarantees with routing performance at different level. It is verified by our simulation that the performance of "anonymous-only ANODR" is comparable to common on demand routing schemes currently in use (e.g., AODV). In addition, we also implement untraceable packet flow so that passive adversaries cannot trace back to the source or the destination. ANODR pays reasonable cost, such as neighborhood traffic mixing, to meet this privacy demand. It is verified by our simulation that the performance of untraceable ANODRs ("anonymous+untraceable ANODR" and "anonymous+untraceable ANODR-KPS") is more efficient than its counterpart designed for wired networks (e.g., MIX-Net).

To defend active attacks, in Chapter 7 we design and simulate *partial trust community* in mobile wireless networks. In the presence of internal adversary, *partial trust* is a fundamental problem in self-organized ad hoc networks. Each ad hoc node must make its decisions on whether to trust its ad hoc peers, and how to regulate the conferred trust. Partial trust requires that a network service to be securely distributed to a community. At the level of each individual node, the service provisioning is untrustworthy and is allowed to be disrupted. However, at the level of community, the service provisioning becomes trustworthy—even if some of community members are selfish or malicious, the service remains available and reliable. Our simulation study of two community-based secure routing protocols, namely community-based AODV and community-based ANODR, verifies the effectiveness of the partial trust community design.

The rest of the dissertation is organized as follows. In Chapter 2 we define a set of anonymity concepts for mobile wireless networks. We seek to illustrate the difference between mobile wireless anonymity research and fixed wired anonymity research. In Chapter 3, we describe our adversary model and the impact of internal adversary in mobile wireless network security design. In Chapter 5 we propose a practical and computationally secure scheme for single-hop data forwarding. We then present anonymous routing attacks in Chapter 4 to demonstrate the feasibility of anonymous routing attacks, thus justify the need of anonymous routing protocols. Our proposed anonymity models and countermeasures are described in details in Chapter 6 and Chapter 7. The difference between Chapter 6 and Chapter 7 is the former one studies countermeasures against passive attacks, and the latter one studies countermeasures against active attacks. In both studies we have extensively evaluated related routing performance using simulation study. Chapter 8 described related anonymity research ad hoc network security research. Finally we summarize the dissertation in Chapter 9.

CHAPTER 2

Problem definition

Our nature consists in motion;
complete rest is death.

–Blaise Pascal

In this chapter we seek to differentiate anonymity research in mobile wireless networks from its counterpart in fixed wired networks. We first describe a set of common definitions used in fixed wired networks, then we point out that it is necessary to modify and extend these definitions before they can be applied in mobile wireless networks. We propose the concept of *mobile anonymity* as our answer. Mobile anonymity defines a set of concepts which is the design goal of anonymous communication in mobile wireless networks.

2.1 Existing anonymity nomenclature in fixed networks

At first, let's briefly overview the nomenclature introduced earlier by Pfitzmann and Köhntopp [111]. In a distributed system or computer network, members are identified by unique IDs. Network transmissions are treated as the *items of interest* (IOIs). *Pseudonym* is an identifier of subjects to be protected. It could be associated with a sender node, a recipient node, or any protégé demanding protection. The concept of *pseudonymity* is defined as the use of pseudonyms as IDs. The concept of *anonymity*

is defined as the state of being not identifiable with a set of subjects, namely the *anonymity set*. In a network comprised of peer nodes, the anonymity set is the set of all (uncompromised) peer members in the network.

The concept of *anonymity* is defined in terms of *unlinkability* or *unobservability*. The difference between unlinkability and unobservability is whether security protection covers IOIs or not:

- *Unlinkability*: Anonymity in terms of unlinkability is defined as unlinkability of an IOI and a pseudonym. An anonymous IOI is not linkable to any pseudonym, and an anonymous pseudonym is not linkable to any IOI. More specifically, *sender anonymity* means that a particular transmission is not linkable to any sender's pseudonym, and any transmission is not linkable to a particular sender's pseudonym. *Recipient anonymity* is similarly defined.

  A property weaker than these two cases is *relationship anonymity* where two or more identity pseudonyms are unlinkable. In particular for senders and recipients, it is not possible to trace who communicates with whom, though it may be possible to trace who is the sender, or who is the recipient. In other words, sender's pseudonym and recipient's pseudonym (or recipients' pseudonyms in case of multicast) are unlinkable.

- *Unobservability*: Unobservability also protects IOIs from being exposed. That is, the message transmission is not discernible from random noise. More specifically, *sender unobservability* means that a could-be sender's transmission is not noticeable. *Recipient unobservability* means that a could-be recipient's transmission is not noticeable. *Relationship observability* means that it is not noticeable whether anything is sent from a set of could-be senders to a set of could-be recipients.

Unobservability states that a transmission event is *not* interceptable by adversary. This can be achieved either by (1) making network transmissions indistinguishable from random physical noises, or (2) maintaining radio silence. The first method may be useful to fool external adversary. But in the presence of internal adversary, anonymity in terms of unobservability can only be achieved by radio silence (or equivalence) so that nobody can receive it. Throughout the paper, we use the term "anonymity" as a synonym of "anonymity in terms of unlinkability" unless explicitly specified.

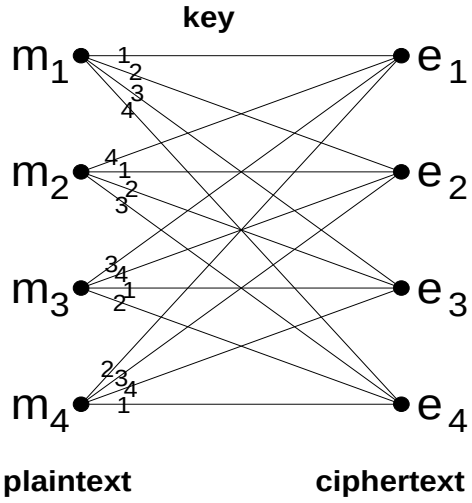

Figure 2.1: **Perfect cipher** $f(m_i, key) \mapsto e_j$ (keys are denoted as numbers)

	e_1	e_2	e_3	e_4
m_1	1	2	3	4
m_2	4	1	2	3
m_3	3	4	1	2
m_4	2	3	4	1

Figure 2.2: **Latin Square representation of Fig. 2.1**

2.2 Problem study: perfect anonymity

What is the upper bound of anonymity protection against content analysis and timing analysis[1]? This is a simple question, but surprisingly with no general answer so far.

[1] In timing analysis, adversary can use temporal causality between successive transmissions to trace a victim message's forwarding path. A packet forwarded to node X at time t and a packet forwarded from the same node at time $(t+\epsilon)$ are very likely on the same packet flow. Any traffic analysis technique using temporal information is in general also timing analysis.

Some equivalent definitions of perfect anonymity require the size of anonymity set to be infinite [133]. In this dissertation we adopt another approach — perfect anonymity can be achieved in a system with anonymity sets of finite size. The concept of perfect anonymity in information theoretic models is similar to the concept of perfect secrecy proposed in Shannon's information theoretic secrecy paper [132]. In addition, this dissertation will introduce a novel concept "time interval policy" to show that timing analysis is indeed solvable in theory.

Shannon developed the notion of *perfect secrecy* for message encryption based on information theory. Perfect cipher is a mathematic relation among three random variables K, M, E in *finite* key space $\mathcal{K}$, message space $\mathcal{M}$, and cryptogram space $\mathcal{E}$, respectively. $H(M)$ denotes the entropy of M. $H(M|E)$ denotes the entropy of M after cryptograms are intercepted by an external adversary. The entropy difference $A(M, E) = H(M) - H(M|E)$ is the amount of information about M which an external adversary obtains.

Intuitively, the adversary gains zero information about message in a perfect system even if it intercepts all cryptograms, i.e., $H(M) = H(M|E)$ (Figure 2.1). This implies $H(M) = H(M|K)$ (Figure 2.3). Given any cryptogram $e \in \mathcal{E}$, an adversary has to uniformly choose the candidate message m from the entire message space $\mathcal{M}$ if the secret key k is uniformly distributed over the key space $\mathcal{K}$. Hence each candidate message is equally likely. As a result, although an adversary knows the finite spaces *a priori* and intercepts all cryptograms, information gained *a posteriori* is no more than the *a priori* knowledge. Any adversary cannot compromise perfect message secrecy even if it is given infinite time to exhaustively search the entire finite spaces. It was shown that *one-time pad* (or *Vernam cipher* [140] with one-time key bits) achieves perfect secrecy as long as the number of keys is not less than the number of messages.

Similarly, ideal anonymity can be defined on uniform distributions and the differ-

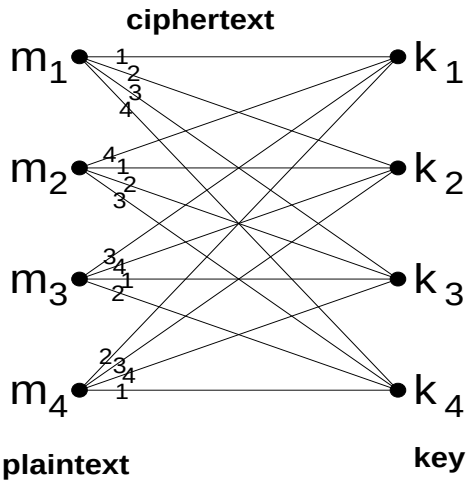

Figure 2.3: **Another statement of perfect cipher:** $H(M) = H(M|K)$ (ciphertexts are denoted as numbers)

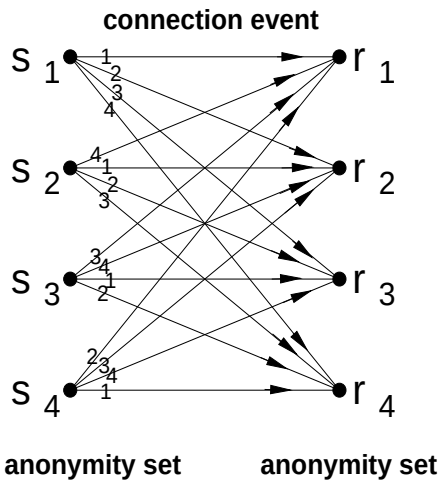

Figure 2.4: **Perfect anonymity when** $N = 4$ (End-to-end connection events are denoted as numbers)

ence between *a priori* and *a posteriori* knowledge. In a computer network, the network size is denoted as N (i.e., there are at most N nodes in the network). Node i is identified by its network address ID_i which is in a finite identity space $\mathcal{I}$ represented as a finite data field of $n = \lceil \log_2 N \rceil$ bits. For example, in IPv4 $n = 32$, and in IPv6 $n = 128$. An adversary knows this design *a priori*.

As depicted in Figure 2.4, simply by replacing plaintext/key with the anonymity set, and ciphertext with end-to-end connection event, we have the upperbound of anonymity protection, namely *perfect anonymity*, in a system of finite members at each moment. As Figure 2.5 and 2.6 illustrate, it is clear that **our design has an information theoretic goal, but needs computational cryptography and network-based mechanisms to realize the goal**. In particular, the information theoretic goal is perfect anonymity; cryptographic tools must be used to hide real transmission events in all transmission events by indistinguishability[2]; and network-based mechanisms include

[2]For two comparable bit strings, the term "indistinguishability" refers to the same concept in modern

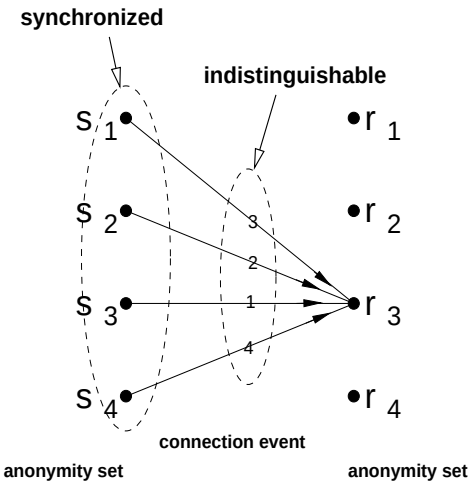

Figure 2.5: **Perfect sender anonymity: synchronized senders & real events indistinguishable from dummy/decoy events** (using an explicit recipient r_3 in example)

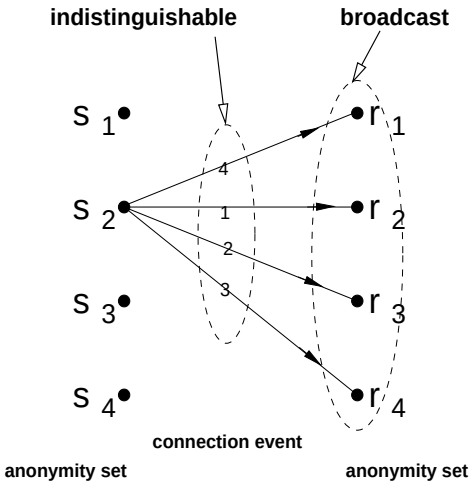

Figure 2.6: **Perfect recipient anonymity: broadcast to all recipients & real events indistinguishable from dummy/decoy events** (using an explicit sender s_2 in example)

broadcast (to all recipients) and synchronization (at sender's side). It is clear that, **in perfect anonymity, a "brute-force" form of synchronization/broadcast that covers the entire anonymity set is an indispensable necessary condition**. The remaining questions are: (1) how to implement this brute-force synchronization/broadcast in an efficient way; and (2) how to trade security with performance if such brute-force efforts are impractical.

For huge anonymity sets (e.g. 2^{32} IPv4 addresses, 2^{48} link layer MAC addresses, and 2^{128} IPv6 addresses), it is infeasible to implement brute-force synchronization/-broadcast. We will employ an information hiding approach — *the information related to these huge anonymity sets must not be used at all in network communication*. This is

cryptography [54]. Ensemble X is indistinguishable from ensemble Y if a $\mathcal{BPP}$ adversary cannot differentiate them with non-negligible probability in polynomial time.

an equivalence of radio silience in terms of information related to such huge anonymity sets.

Along the timeline, we will propose "*time interval policy*" to expand such perfect anonymity over the temporal dimension. The time interval policy seeks to realize the momentary perfect anonymity at each time interval of a pre-defined length. During each time interval, the system must either maintain radio silence, or if an end-to-end connection event ever happens, then the scenario depicted in Figure 2.4 must be realized during that interval. In Chapter 6 we will propose ideal models to realize these concepts after we define them below.

2.3 Problem study: anonymity in mobile networks

In this dissertation we consider unicast communications[3] between a sender and a recipient. Moreover, we limit our research scope in anonymous data forwarding and anonymous routing. In other words, we do *not* seek to cover untraceability problems at the physical layer or the application layer. For instance, it is beyond the scope of this dissertation to study how to trace a network node using signal delay and triangulation at the physical layer, or to study how to trace a user application based on the application's idiosyncratic communication pattern.

We first treat the underlying network as a homogeneous peer-to-peer network with dynamic membership. There is no hierarchical relation between any network members. In other words, all network members are peers. They may join or leave the network at their own will. Real world examples of peer-to-peer networking include P2P networks where network members are connected by the Internet cloud, and ad hoc networks where network members are connected by wireless radio.

[3] Multicast, anycast, and manycast communications are future work to be addressed.

This design choice gives us a simple yet general anonymous networking model. The underlying network is an undirected graph $G = \langle V, E \rangle$, where peer nodes form a vertex set V, and communication links form an edge set E. Other problems, such as anonymous routing in a hierarchical graph, or anonymous transaction between user applications and nodes, etc., can be regarded as supplementary problems built on top of such a peer-to-peer anonymous network.

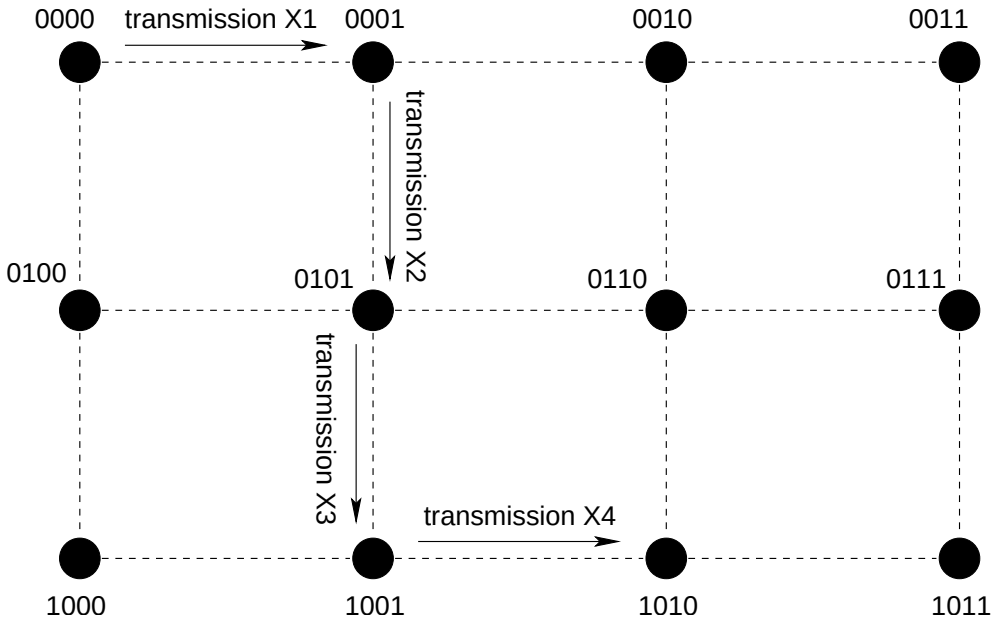

Figure 2.7: **An end-to-end connection event in a simple peer-to-peer network**

Then we introduce mobility and network dynamics into the simple network model.

- As nodes randomly move, the relation between a network node and a vertex of the underlying graph is broken. After mobility is introduced, vertexes in the underlying graph represent an array of adversarial traffic analysts who are monitoring nearby wireless traffic. As depicted in Figure 2.8, within each traffic analyst's eavesdropping range, it can correlate a mobile node with its current venue/vertex. The analysts can divide the network into eavesdropping "cell"

(e.g., using Voronoi diagram), each cell corresponds to a vertex in the underlying graph G. In other words, now a vertex in the underlying graph may hold any set of network nodes at any time.

- Nodes may not be available due to selfishness, system crash, and energy exhaustion, etc. Thus a node's online probability $p_{on} \leq 1$. If a node is not online, it disappears from the graph and all associated links also disappear. However, when the node decides to be online again, the node and all associated links are restored immediately.

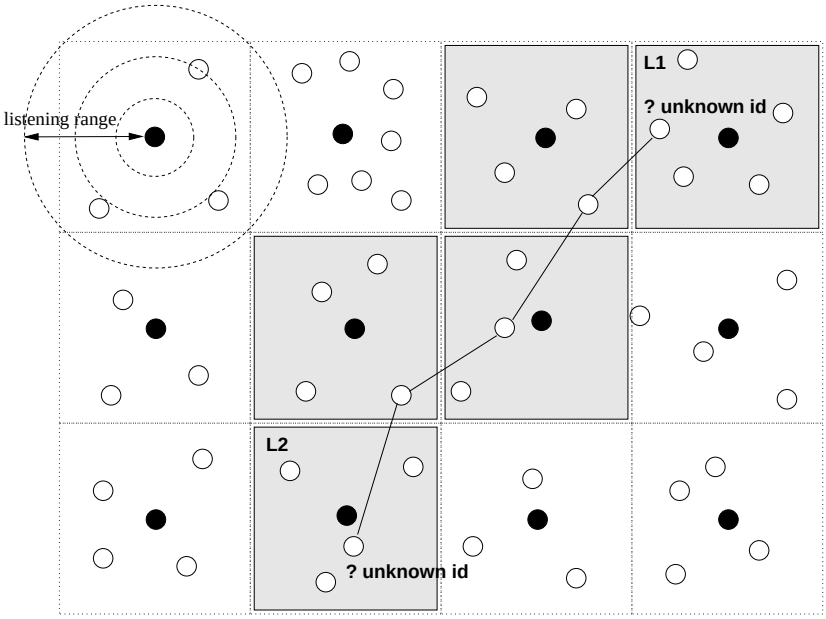

Figure 2.8: **Identity anonymity vs. venue anonymity in mobile networks** (Traffic analysts are depicted as solid black nodes. Identified active routing areas are depicted in shade.)

These two assumptions delineate the network topology at a specific moment and mobile changes over the timeline. Now we can give an intuitive version of anonymity definitions in a mobile network of peers. In the network an end-to-end connection event along a (multi-hop) path is comprised of consecutive transmission events. For

example, the end-to-end connection event depicted in Figure 2.7 is comprised of four consecutive transmission events $\langle X_1, X_2, X_3, X_4 \rangle$. Assuming all transmission events are interceptable, the network needs following anonymity supports:

- **Sender anonymity**: *Sender anonymity* means an adversary cannot correlate the sender's identity or its venue with an actual (multi-hop) end-to-end connection event. In other words, sender anonymity in mobile networks is comprised of two aspects: *sender identity anonymity* and *sender venue anonymity*.

  Here the term "venue" means a specific vertex position in the underlying graph G. It is important to note that, in fixed networks, a sender/recipient and its venue are synonyms—identifying a sender/recipient's venue implies the compromise of sender/recipient anonymity. In mobile networks, this is not true because a node's identity is dissociated with its venue. Therefore, *identity anonymity* and *venue anonymity* in mobile networks are different concepts—breaking identity anonymity does not imply breaking venue anonymity, and vice versa.

  For example, in Figure 2.8 an adversary may be able to identify a mobile sender's venue/vertex $L1$, but fails to identify the sender's identity. Another example is in DSR [73] or AODV [108], any route request forwarder knows the sender's identity, but may not know the venue/vertex of the sender.

- **Recipient anonymity**: The two aspects of *recipient anonymity*, namely *recipient identity anonymity* and *recipient venue anonymity*, are similarly defined. In Figure 2.8, an adversary may be able to identify a mobile recipient's venue $L2$, but fails to identify the recipient's identity. Another example is in DSR [73] or AODV [108], any route request forwarder knows the recipient's identity, but may not know the venue of the recipient.

- **Sender-recipient relationship anonymity**: Sender-recipient relationship ano-

nymity means the adversary cannot identify the correlation between a sender and a recipient. In particular, along a (multi-hop) path this means the adversary cannot trace a packet flow towards the sender/source or the recipient/destination.

- **Location privacy**: Unlike fixed networks, location privacy supports are needed in mobile networks. *Location privacy* means an adversary cannot identify network members and their communication patterns in a local neighborhood. Neighborhood is defined in terms of hop count in the underlying graph G. In particular, one-hop neighborhood of a vertex covers its eavesdropping range.

- **Motion pattern privacy**: Like location privacy, *motion pattern privacy* is a new anonymity aspect in mobile networks. It means an adversary cannot identify where a set of network members were and are, as well as any indication between the places where the network members were and are (e.g., "node A is moving towards east!").

These notions are different from the notion of anonymity in user transactions (e.g., digital cash [30]), which is normally implemented by pure cryptographic protocols using zero-knowledge proof (ZKP) [56]. By simply presenting pseudonyms to a ZKP verifier, a ZKP transaction effectively hides its sender's real ID, or its recipient's real ID, or the relation between the pair of real IDs. This means sender, recipient, and relationship anonymity is achieved by pseudonymous transactions using cryptographic ZKP.

The above notions are very different. As depicted in Figure 2.7, vertexes are given unique pseudonyms. Let's assume no extra anonymous protection is provided except a pseudonymous system. It is very important to note that the adversary can name every vertex using its own pseudonym system. If a node at vertex 0000 (in the adversary's pseudonym system) sends a message to a node at vertex 1010 via vertex

set $\{0001, 0101, 1001\}$, then anonymity in terms of unlinkability is compromised after the adversary intercepts all transmissions $\{X_1, X_2, X_3, X_4\}$ and sees the same message contents in both transmissions. Using its own pseudonym system, the adversary conforming to our adversary model can successfully identify vertex 0000 and vertex 1010 as the actual sender's venue and recipient's venue, respectively. Clearly in this example, a simple pseudonymous system cannot ensure anonymity. Pseudonymity is merely a necessary but not sufficient condition of anonymity studied in this work.

To formalize the intuitive notions described above, now we name the set of these notions as "mobile anonymity" and give its definition.

2.4 Mobile anonymity: the formal notion

We first formally define the concepts of transmission event and end-to-end connection event. As we stated earlier, this dissertation will not address physical layer signal tracing or triangulation. The definitions given below only consider issues related to forwarding and routing protocol design.

Definition 1 *A **transmission event** is an interceptable packet in its known packet format, with its contents, its interception time, and its interception venue/vertex recorded.*

*An **end-to-end connection event** is a collection of transmission events between a sender and a recipient.* □

The following definition defines the meaning of $\rightsquigarrow$ to quantify the degree of anonymity guarantee.

Definition 2 *Anonymity measurement entropy $H(A) \rightsquigarrow H(B)$ if the degradation difference $(H(B) - H(A)) < c$ where $H(B)$ is an entropy bound and c is a system-*

defined constant that the network's security policy allows. If $c = 0$, then we say the related anonymity support is ***perfect****.* □

The formal notion of mobile anonymity is defined in terms of uncertainty entropy and operator $\rightsquigarrow$:

Definition 3 ***(Mobile anonymity)*** *Given a network sender s in a known identity space $\mathcal{I}$ of finite size N, an adversary's knowledge about this sender before any transmission is the uncertainty entropy $H(I_s) = H(I) = \log_2 N$. Let transmission space $\mathcal{X}$ be the set of all interceptable network transmissions, and X be a random variable of $\mathcal{X}$. $H(I_s|X)$ is the adversary's knowledge about the sender after intercepting all transmissions. The network ensures* ***sender identity anonymity*** *if $H(I_s|X) \rightsquigarrow H(I)$.*

Given an underlying graph $G = \langle V, E \rangle$, by replacing identity space $\mathcal{I}$ with the vertex set V of the underlying graph G, the network ensures ***sender venue anonymity*** *if $H(V_s|X) \rightsquigarrow H(V)$.*

Given a network recipient r, ***recipient identity anonymity*** *and* ***recipient venue anonymity*** *are similarly defined as $H(I_r|X) \rightsquigarrow H(I)$ and $H(V_r|X) \rightsquigarrow H(V)$.*

The network ensures ***sender-recipient identity relationship anonymity*** *if the relationship entropy between s and r, $H((I_s, I_r)|X) \rightsquigarrow H(I, I)$, which is $2 \cdot \log_2 N$ when the sender and recipient are independently chosen. The network ensures* ***sender-recipient venue relationship anonymity*** *if any two tranmission events of the same end-to-end connection event (packet flow) cannot be correlated, that is, they are indistinguishable from two independent random transmission events by a $\mathcal{BPP}$ adversary.*

In a mobile network, we assume a vertex of the underlying graph knows its exact location (e.g., via GPS). ***Location privacy*** *or* ***strong location privacy*** *means two properties: (1) A sender/recipient within a vertex's eavesdropping range can be any node from the identity space $\mathcal{I}$, i.e., $H(I_s) \rightsquigarrow H(I)$ and $H(I_r) \rightsquigarrow H(I)$; (2) Any two*

transmission events within a vertex's eavesdropping range, even happened at different time, cannot be correlated together. That is, they are indistinguishable from two independent random transmission events by a $\mathcal{BPP}$ adversary. If an anonymous scheme can ensure (1) but not (2), then we say that ***weak location privacy*** *is ensured.*

Motion pattern privacy *or* ***strong motion pattern privacy*** *means two properties: (1) A sender/recipient of any (remote) transmission event can be any node from the identity space $\mathcal{I}$, i.e., $H(I_s) \rightsquigarrow H(I)$ and $H(I_r) \rightsquigarrow H(I)$; (2) Any two transmission events, either of a single node or of a group of nodes, and even happened at different time, cannot be correlated together globally. That is, they are indistinguishable from two independent random events by a $\mathcal{BPP}$ adversary. If an anonymous scheme can ensure (1) but not (2), then we say that* ***weak motion pattern privacy*** *is ensured.* □

Weak location privacy and weak motion pattern privacy state that a node's location at each moment and the node's motion pattern (i.e., locations at various moments) are not identifiable given the node's identity. Strong location privacy and strong motion pattern privacy state that even any communication pattern is not revealed to the adversary. The term "two transmission events cannot be correlated" (or "two transmission events are not correlatable") means either they indeed independently occur, or even if their occurrences are related, the adversary has to identify the relation by inverting one-way functions (or differentiating cryptographically strong pseudorandom bits from truly random bits).

CHAPTER 3

Adversary Model

The true enemy is the enemy within.

–Buddhism quote

3.1 Threats to mobile wireless networks

The nature of mobile wireless networks makes them very vulnerable to an adversary's malicious attacks. We classify threats to mobile wireless networks into three major categories: (1) wireless link intrusion, (2) mobile node intrusion, and (3) algorithm exploitation. They are described in more details below.

First, the use of wireless links renders any wireless network susceptible to attacks ranging from passive eavesdropping to active interfering. Unlike wired networks where an adversary must gain physical access to the network wires or pass through several lines of defense at firewalls and gateways, an wireless adversary can easily launch attack once it is within radio transmission range.

Second, mobile nodes are autonomous units that are capable of roaming independently. This means that nodes with inadequate physical protection are receptive to being captured, compromised, and hijacked. Since tracking down a particular mobile node in a large scale wireless network cannot be done easily, attacks by a compromised node from within the network are far more damaging and much harder to detect. There-

fore, any node in a mobile wireless network must be prepared to operate in a mode that trusts no single peer.

Third, decision-making in many mobile wireless networks, in particular multi-hop ad hoc networks, is usually decentralized or fully distributed. Many network algorithms rely on the cooperative participation of all mobile nodes. The lack of centralized authority means that the adversaries can exploit this vulnerability for new types of attacks designed to break the cooperative algorithms.

The three classes of threats can be combined. A real world threat is normally a complex combination of all these three classes. This means a sound network security protocol should resist (1) wireless link attackers and (2) mobile node attackers using (3) *provably secure* algorithms/protocols to protect the network. If a provably secure protocol incurs excessive overheads, we need to show how to trade security protection with performance.

3.2 Adversary model

In general, adversary can be classified into two major categories according to various criteria: *external* adversary or *internal* adversary according to its network membership status; and *passive* adversary or *active* adversary according to its behavior.

In a computer network, an *external adversary* is a *link* intruder that only poses threat to all links. We allow an external adversary to intercept *all* traffic transmitted on all the connections in the network. An *internal adversary* is a *node* intruder that can compromise legitimate network members. We further specify our passive and active adversary models with respect to network membership.

3.2.1 Passive adversary model

A *passive external adversary* knows and actualizes all network protocols and functions. It is a $\mathcal{BPP}$ adversary (ref. Appendix A.1) who can intercept and record all eavesdropped traffic. However, legitimate network members can employ public key cryptosystems (e.g., RSA, El Gamal) and symmetric key cryptosystems (e.g., AES, SHA1) to protect critical messages. With *bounded computing time*, passive external adversary cannot break these well-defined cryptosystems with non-negligible probability. In particular, it cannot effectively invert one-way functions, or differentiate cryptographically strong pseudorandom ensembles (CSPRE) from truly random ensembles.

For cryptosystems used in this work, our major concern is *performance*. Compared to symmetric key operations, public key processing on resource-limited nodes is relatively much more expensive. Related publications [23] [60] [78] have reported measured computational delay of public key crypto-operations on heterogeneous platforms. For reasonable key size (e.g., 1024-bit for RSA and El Gamal) and portable mobile devices (e.g., iPAQ pocket PC and Palm Pilot), the computation delay is at least hundreds of milliseconds, and sometimes several seconds. On the contrary, symmetric key crypto-operations only incur computational delay measured in microseconds on the same devices and on application data of the same size. Additionally, even though this work does not study sensor network security, it is possible that our mobile nodes need to communicate with a distributed sensor network on the move. Sensor nodes have very limited capability of both computing and communication. Public-key cryptosystems are not only expensive, but also hardly available on sensor nodes. Due to these practical reasons, in this work *we seek to avoid the use of public key cryptography in our practical protocol design* (but not in theoretic models).

In addition, we also consider *passive internal adversary*. In ad hoc networks,

nodes with inadequate physical protection are susceptible to being captured and compromised. Tamper resistance based countermeasures could be effective against node compromise. However, the related research is addressing physical protections. As a network security research work, we will not consider this design choice. Instead, we assume that *all* private records on a victim node, including all private keys and private route caches, are revealed if the node is compromised. In addition, since there is no intrusion detection guarantee in ad hoc networks [147][92], internal adversary can stay in the network for a considerable period until it is detected, identified, and excommunicated (at that time obviously it stops to be an *internal* adversary). In summary, passive internal adversary is characterized by: (1) After the adversary compromises a victim node, it can see the victim's currently stored records including private keys, inputs, random bits, and route caches. (2) The adversary may move from one node to another over time (i.e., *mobile adversary* proposed in previous research [63]). However, its capability to intrude legitimate members is not unbounded. During a time window T_{state} it cannot compromise more than c members. (3) We argue that *any intrusion detection system is not perfect*. A passive internal adversary exhibiting no malicious behavior will stay in the system and intercept all messages forwarded to it. This means that simple encryption cannot stop an internal adversarial forwarder who is granted the privilege to decrypt routing messages.

Internal adversary presents great challenge to mobile anonymity design. We observe that common intrusion detection and neighborhood discovery schemes are *incompatible* with anonymous routing if internal adversary is considered. Common intrusion detection systems require all network members to identify themselves so that a centralized or distributed detection algorithm can monitor and block suspected nodes [147]. Similar to this argument, a neighborhood discovery protocol also requires all mobile nodes to reveal their identities to their neighbors. Unfortunately, any *internal* adversary can always break location privacy support using such intrusion

detection or neighborhood discovery mechanism. In a nutshell, the combination of internal adversary and intrusion detection (or neighborhood discovery) directly conflicts with anonymity and location privacy requirements. As location privacy is considered as one of the application demands, *we would avoid the use of intrusion detection and neighborhood discovery schemes*. Instead, we adopt an intrusion tolerant approach in answering the challenge imposed by internal adversary.

3.2.2 Active adversary model

Like passive adversary, active adversary is also classified into two types: active *external* adversary and active *internal* adversary. An active external adversary is not a legitimate network member, thus it can be effectively isolated by applying cryptographic countermeasures, e.g., encryption and message authentication. In particular for multi-hop ad hoc routing, TESLA [110] is used in Ariadne [65] (a secured version of DSR [73]); while certified digital signature is used in ARAN [125] (a secured version of AODV [108]) to defend active external adversary.

Unfortunately, an active *internal* adversary is a compromised network member, we must devise new countermeasures to answer the challenge. Intrusion detection is a candidate solution. But as we stated previously, it is incompatible with anonymity and location privacy design goals. In Chapter 7 we will present an intrusion tolerant based countermeasure, namely “partial-trust community”, to defend active internal adversary.

3.3 Routing attacks based on adversary model

In previous ad hoc network security research, the routing attacks can be roughly classified into two major categories: (1) Active attacks [65] [125], where adversarial nodes

seek to actively disrupt route discovery or data delivery procedures; (2) Passive attacks [79], where adversarial nodes seek to passively evaluate network characteristics using traffic analytical techniques, but without introducing obvious anomalies. Active attack can be further classified into more sub-categories:

- Rushing attack [67] and wormhole attack [66] are meta-attacks against route discovery. By these attacks adversarial nodes can significantly increase their chances to be placed on ad hoc routes.

- After the route discovery procedure, if some adversarial nodes are selected to be on an ad hoc route, then they can launch "black hole" attack to drop all packets, or launch "gray hole" attack to selectively drop some packets. These attacks seek to directly disrupt data forwarding service.

- There are various forms of "denial-of-service" attacks. Instead of participating in data forwarding, such attackers may simply try to deny ad hoc routing services by breaking the cooperative routing algorithms. Examples of such attacks include sending false route error (RtERR) report, lying about routing metrics (e.g., hop count in AODV, forwarder list in DSR), creating loops on multi-hop path, poisoning other nodes' routing cache, etc.

In this work, we also classify routing attacks into another two major categories based on the adversary model: *external attacks* and *internal attacks*. This classification is not new. Lakshmi and Agrawal [84] briefly described this classification, but did not address internal attack in their proposal. By this classification, external attacks can be addressed by cryptographic methods. For example, either TESLA (as in Ariadne [65]) or digital certification (as in ARAN [125]) can be used to defend active attacks launched by external attackers.

However, internal attacks are very different. A similar differentiation is described by Hu et al. [65] as "Active-y-x" where there are y internal adversarial nodes and x external adversarial nodes in the entire network. Because cryptographic authentication cannot effectively differentiate legitimate members from internal adversaries, the y internal adversaries can disrupt ad hoc routing as if there is no cryptographic protection. Therefore, network-based solutions must be devised to answer the challenge. In later chapters, namely Chapter 5, Chapter 6, and Chapter 7 we will incrementally propose a series of solutions to defend external and internal attacks in the context of (1) single-hop data forwarding, (2) multi-hop routing against passive attacks, and (3) multi-hop routing against active attacks, respectively.

CHAPTER 4

Anonymous Routing Attacks

Strength lies not in defense but in attack.

–Fuhrer of the Third Reich

In this chapter we demonstrate that both existing anonymity schemes designed for fixed networks and existing ad hoc routing schemes do not provide mobile anonymity protection in mobile wireless networks,. We conclude that new anonymous routing schemes must be devised to answer the challenge.

4.1 New anonymity threats in mobile wireless networks

First we refer to Fig. 2.8 to show the referential case for collaborative adversaries to trace the motion pattern of a mobile node. A collection of adversarial nodes can be (pre-)deployed to cover a region or even the entire ad hoc network. As depicted in Figure 2.8, the adversaries can divide the network into cells based on radio receiving range. One or more adversarial nodes can effectively monitor each cell. Any open wireless transmission within one-hop transmission range is thus collected and fed back to adversary's computing center for further analysis. The examples described below demonstrate various passive attacks that can be launched by the adversaries. Later in this chapter we will show more advanced attacks that only require sparse adversary.

4.1.1 Differentiation between identity anonymity and venue anonymity

In fixed networks, a sender (or recipient) and its venue are synonyms, that is, identifying a sender's (or recipient's) venue implies the compromise of sender (or recipient) anonymity. In mobile networks, this is not true because a node's identity is dissociated with a specific venue. Here the term "venue" means an identifiable location in the network. In Chapter 2 we defined the term "venue" by modeling the network as a simple underlying graph G.

Example 1, Example 2 and Example 3 use Figure 2.8 to show that *identity anonymity* and *venue anonymity* in mobile networks are different concepts. In other words, breaking identity anonymity does not imply breaking venue anonymity, and vice versa.

Example 1 (Sender or recipient identity anonymity attack in on-demand route request flooding) *In common on-demand ad hoc routing schemes like DSR [73] and AODV [108], identities of the source/sender and the destination/recipient are explicitly embedded in route request (RtREQ) packets. Any external adversary who has intercepted such a flooded packet can uniquely identify the sender's and the recipient's identities, but may not know the venue/vertex of the sender or the recipient.* □

Example 2 (Per-hop encryption may not protect sender or recipient identity anonymity against internal adversary) *A seemingly-ideal cryptographic protection is to apply pairwise key agreement on every single hop, so that a single-hop transmission is protected by an ideal point-to-point secure channel between the two ends of the hop. The secure channel protects an entire packet including the packet header.*

This solution prevents external adversary from understanding routing messages and network topology, but unfortunately does not prevent any internal DSR/AODV network member from identifying the sender's and the recipient's identities upon receiving an RtREQ packet. □

Example 3 (Packet flow tracing attack) *Packet flow tracing attacks reveal the relationship between a sender's venue and its recipient's venue. On a (multi-hop) forwarding path, adversary can use timing correlation and content correlation analysis to trace a packet flow.*

- Timing correlation analysis: *The adversary can use timing information between successive transmission events to trace a victim message's forwarding path. With no background traffic, a packet forwarded to node X at time t and a packet forwarded from the same node at time $(t + \epsilon)$ are very likely on the same packet flow.*

- Content correlation analysis: *A control/data flow can be traced by content correlation (e.g., comparison on data field contents and length).*

In Figure 2.8, collaborative adversarial analysts can trace an ongoing packet flow to the sender's venue $L1$ and the recipient's venue $L2$, thus break sender (or recipient) venue anonymity. But they may not be able to identify the sender's identity. This is possible in ANODR [78] where sender's and recipient's identities are not used in on-demand route discovery packets. □

4.1.2 Privacy of network topology

In fixed networks, routing does not affect network topology which is physically determined *a priori*. However, this is not true in mobile networks where network topology constantly changes due to mobility. Once information about the network topology is revealed, the adversary can break the network's anonymity protection given other out-of-band information like geographic positions and physical boundaries of the underlying mobile network. Privacy of network topology becomes a new anonymity aspect in mobile networks.

In fixed Internet, proactive routing schemes like BGP [120], OSPF [96] and RIP [62] are widely used in inter-domain routing and intra-domain routing. Every router possesses abundant knowledge about network topology if the underlying routing scheme is hierarchical, or complete knowledge about the entire network topology if the underlying routing scheme is flat. This does not affect anonymity protection in fixed networks because network topology is already physically determined *a priori*. In proactive ad hoc routing protocols like DSDV [107], OLSR [3] and TBRPF[101]), mobile nodes also constantly exchange routing messages, so that each sender node knows enough network topological information to find any intended recipient. In a typical network with arbitrary pairwise connection pattern, this means at each moment every sender node knows abundant network topological information about all other nodes. Thus a single adversarial sender can break anonymity protection of the underlying mobile network. This remark is justified in the following Example 4 and Example 5.

Example 4 (A compromised sender tries to locate where a specific node is) *An anonymous routing protocol should prevent a sender from knowing a (multi-hop) forwarding path towards any specific mobile node. Otherwise, a compromised network member can simply function as a sender to trace any mobile node at its convenience. This example shows that pre-computed routing schemes, in particular proactive routing schemes that accumulate a priori network topology knowledge on each sender, directly conflicts with anonymity protection in mobile networks.*

Any equivalence of proactive routing scheme, such as enforcing requirement to let node send out unsolicited advertisements to other nodes so that network topology can be well-known in the network, also directly conflicts with mobile anonymity protection. The network topology knowledge collected on mobile nodes can be used by the adversary to fight against the network. If node compromise is feasible, such design indeed establishes a lot of single points of compromise in the network. □

MIX-Net [28] is a common anonymous communication scheme widely used in the fixed networks. In MIX-Net, the entire forwarding path must be determined on the sender prior to anonymous data delivery. Proactive routing schemes allow a sender to make the decision, but this design choice is not resilient to internal threats.

Example 5 (Vulnerabilities of MIX-Net in mobile networks) *In Chaumian MIX-Net [28], each sender MIX must pre-compute its routing path towards its recipient. Various patches have been proposed to fix security problems in Chaum's original proposal (e.g., onion length should not increase as hop count increases [105], weakness in public key encryption using raw RSA [114]). Various topological forms of MIX-Net, such as cascade and free route network, have been proposed to implement routing path. Nevertheless, all these MIX-Net variants inherit the same assumption — a sender must pre-compute its routing path before anonymous data delivery. If we directly port Chaumian MIX-Net into a mobile network by treating all or some mobile nodes as Chaumian MIX nodes, then any adversarial sender becomes a single point of compromise.* □

Compared to source routing [115] and link state routing [62] schemes, distance vector routing [96] and virtual circuit [7] based schemes only cache information about immediate next stop for each destination/recipient. This reveals less network topology when a node is compromised. On the other hand, compared to proactive schemes, on-demand schemes are less vulnerable to internal threats since they do *not* require a mobile node to acquire network topology knowledge *a priori*. Based on these observations, we argue that the combination of distance vector (or virtual circuit) based routing schemes and on-demand approach provides better anonymity support in mobile wireless networks.

4.1.3 Privacy of location and motion pattern

In fixed networks, a fixed node's topological location and related physical location are determined *a priori*. Besides, the motion pattern of a fixed node is not a network security concern. In other words, there is no need to ensure privacy for a network node's location and motion pattern. Therefore, in anonymity solutions proposed for fixed networks, a network node is allowed to know its neighborhood. For example, a Chaumian MIX knows its immediate upstream and downstream MIXes, a jondo in Crowds [119] knows its next jondo or the destination recipient. If directly ported from the fixed networks, these schemes do not ensure location privacy near any internal adversary, which can launch attacks described in Example 6.

Example 6 (One-hop location privacy attack) *Given any cell L depicted in Figure 2.8, the inside wireless traffic analyst may gather and quantify (approximate) information about active mobile nodes, for example, (a) enumerate the set of active nodes in L; (b) related quantities such as the size of the set; (c) traffic analysis against L, e.g., how many and what kind of connections in-and-out the cell.* □

Ensuring privacy for mobile nodes' motion pattern is a new expression. Example 7 gives a brief overview of the attack. If the network fails to ensure one-hop location privacy, Kong et al. [79] showed that a mobile node's motion pattern privacy can be compromised by a dense grid of traffic analysts, or even by a sparse set of internal adversarial nodes under certain conditions, for example, when (1) a node is capable of knowing neighbors' relative positions (clockwise or counter-clockwise), and (2) in DSR or AODV on demand route discovery, RtREP traffic of the same source-destination pair is correlatable.

Example 7 (Motion pattern inference attack) *As implied by the name, the goal of this passive attack is to infer (possibly imprecise) motion pattern of mobile nodes. For*

example, collaborative adversaries can monitor wireless transmissions in and out a specific mobile node, they can combine the intercepted data and trace the motion pattern of the node. In some cases, a network mission may require a set of legitimate nodes to move towards the same direction or a specific spot. Motion pattern inference attack can effectively visualize the outline of the mission. In a network with dense adversarial nodes, motion pattern inference attack can be implemented on top of location privacy attack based on historical records. □

Mobile networks could be deployed in severe environments, where nodes with inadequate physical protection are susceptible to being captured and compromised. Any node in such a network must be prepared to operate in a mode that allows no gullibility. In the network, the combination of location privacy demands and infrastructureless mobile wireless routing schemes presents a dilemma described in Example 8.

Example 8 (Location privacy dilemma in infrastructureless mobile wireless routing schemes) *In mobile wireless routing schemes without infrastructure support, a node must rely on at least one of its neighbors to forward its packets. When anonymity service is concerned, a mobile node is facing a dilemma. On one hand, it must forward its packets to one of its neighbors, so that the neighbor(s) can further forward the packets towards the destination. On the other hand, the node does not know whether there is an adversarial node among its neighbors, and if yes, which neighboring node is adversarial. This dilemma calls for a solution that accomplishes data forwarding without revealing a node's identity information to neighbors.*

4.1.4 Privacy of communication pattern

We can hide a sender/recipient's real identity by using a static pseudonym (e.g., an encrypted real identity or a Zero Knowledge Proof [56] of the real identity). Unfor-

tunately, such a static pseudonym becomes another identity of the node. This naive scheme only hides what the real identity is, but not the characteristics associated with the real identity. As described in Example 9, communication patterns associated with the real identity are revealed as usual.

Example 9 (Communication pattern analysis) *If a static (though encrypted) node pseudonym is assigned to a mobile node, a local adversary can analyze local communication pattern. Intuitively, given arbitrarily x (e.g., $x = 100$) locally intercepted data packets, the adversary may see the relation among these 100 packets. That is, communication pattern is identifiable. The two extreme cases are: (1) all 100 packets were transmitted from one static node pseudonym to another static node pseudonym, and (2) the 100 packets were transmitted from 100 distinct node pseudonyms to 100 distinct node pseudonyms.*

An ideal countermeasure should ensure that any two packets are of identical size, and they look like two random independent transmission events with random packet contents. In other words, given the 100 locally intercepted packets, the two extreme cases and all in-between cases are equally likely to the adversary. □

Based on partial knowledge of the network, the adversary may use communication pattern analysis as a very useful tool to break anonymity protection. For example, if by some means the adversary knows a mobile node corresponds to “Rumsfeld”, and the adversary already knows the communication pattern between the node “Bush” and the node “Rumsfeld”, node “Bush” is then exposed to danger based on the partial knowledge about the network.

We further show more advanced passive attacks to visualize ad hoc routes and mobile nodes’ motion patterns.

4.2 Vulnerability of existing on-demand routing schemes

In proactive ad hoc routing protocols (e.g., DSDV [107], OLSR [3], and TBRPF[101]), nodes constantly exchange routing messages that can be intercepted by passive adversarial nodes. The (internal) adversaries can discover the entire network topology, and can visualize the network topology by finding the location of each transmission node at the granularity of cell. Any equivalence of proactive routing scheme, such as enforcing requirement to let node send out unsolicited advertisements to other nodes so that network topology can be well-known in the network, is also vulnerable to internal attacks. The network topology knowledge accumulated on network members can be used by internal adversary to fight against the network. In the presence of internal adversary, we observe that such design indeed establishes a lot of points of compromise in the network, rather than protecting mobile anonymity for the network.

Compared to proactive schemes, on-demand schemes are less vulnerable since they just set up routes as needed. Nevertheless, common on-demand ad hoc routing protocols are also insecure. Sender/recipient venue anonymity and identity anonymity can be easily compromised by external adversary. Adding per-hop encryption protection is helpful, but it cannot defend internal adversary because route request (RtREQ) packets reveal the source/sender's and the destination/recipient's identities to all network members.

Sender-recipient venue relationship anonymity (or route traceability) is not supported either. A DSR [73] route is traceable since the protocol explicitly embeds routing information in packet headers. From a single intercepted packet, adversaries can know the identity of all forwarding nodes and can visualize the on-demand route at the granularity of cell. AODV [108] is more untraceable because routing information is stored in routing tables instead of packet headers. Nevertheless, it is traceable by collaborative eavesdroppers:

- By inspecting packet header, for example, simply following the forwarding chain in the unprotected link layer header, collaborative eavesdroppers can visualize an AODV route at the granularity of cell. In other words, if a region is covered by multiple collaborative eavesdroppers, then they can visualize all AODV paths intersected with the region. In our adversary model an eavesdropper with unbounded sniff range is assumed, thus all AODV routes can be visualized.

- Even if routing information is ideally encrypted, the adversary can use timing analysis to trace a victim message's forwarding path. A technique called *mixing* [28] can thwart this attack. Such mixing techniques include sending messages in reordered batches, sending dummy packets, and introducing random delays. However, applying such techniques in ad hoc networks may incur significant communication overheads to regular packet forwarding.

Location privacy is not supported. In DSR and AODV, a node knows its neighbors, at least the upstream forwarder and the downstream forwarder of any alive connection. A node can identify local communication patterns, such as counting the number of neighbors and ongoing connections. Motion pattern privacy can be compromised by collaborative location privacy attackers. Later in this chapter we will show a motion pattern privacy attack using correlated RtREP packets and sparse attackers.

4.3 Potential solution: per-hop encryption

One appealing solution is to change [11]'s network-wide key to hop-based link encryption keys. This means each mobile node is allowed to have acquaintance only with its one-hop neighbors, to establish a shared secret key with every neighbor, and to encrypt all exchanged messages with different keys. Then a node intrusion would ideally only compromise the transmissions related to the compromised keys cached

by the victim. Nevertheless, common ad hoc routing schemes secured by this simple solution are vulnerable to anonymity attacks.

1. Some recently proposed secure ad hoc routing protocols, such as SEAD [64], Ariadne [65], and ARAN [125], focus on authentication rather than untraceability. They can be used to stop message injection and modification attacks, but not the passive attacks studied in this work.

2. Sender-recipient venue relationship anonymity (or route untraceability) is partially supported. Content correlation attack is partially stopped if the underlying encryption scheme is sound. However, causality correlation attack is not addressed.

3. Simple encryption does not stop internal adversaries. An internal adversary can decrypt an RtREQ packet and compromise sender/recipient identity anonymity. If an internal adversary is within one hop of the sender/recipient, then it can compromise sender/recipient venue anonymity. As to location privacy, any internal adversary knows the status of its neighborhood.

4. We will present *H-clique* attack, a special motion pattern inference attack, to demonstrate that hop-based link encryption web is not a *sufficient* condition to stop passive internal adversaries — motion pattern privacy can be compromised by passive internal adversaries even though a web of per-link encryption is realized. Hence the problem must be solved by devising new countermeasures where one-hop neighbor information is also protected as privacy.

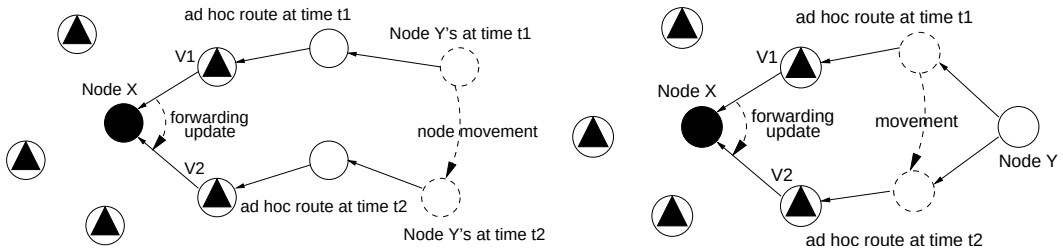

Figure 4.1: **Motion pattern inference attacks** (left: target movement; right: forwarding node movement. Passive internal adversary is depicted as a solid black node in an H-clique. Triangle nodes are the adversary's one-hop neighbors. They are network members but not necessarily adversarial)

4.4 Cumulative H-clique attack

The referential scenario (Figure 2.8) requires *dense* adversarial nodes deployed in the network. It can compromise network nodes' mobile anonymity at the granularity of cell. The H-clique attack only requires *sparse* adversarial nodes. Even though the information gathered by the adversaries is imprecise, they may be able to infer general motion patterns of mobile nodes from ongoing routing events. Figure 4.1 depicts possible motion patterns when an internal adversary X finds its next hop towards a node Y is updated. The underlying attack assumption is that route discovery events from the same pair of source/destination can be correlated together. The update event implies that likely either the target node Y (left figure) or some intermediate forwarding nodes (right figure) have moved along the direction $V_1 \rightarrow V_2$ (clockwisely).

The attack is passive, and it only requires a set of one-hop neighbors. We name such a one-hop set as *H-clique* (i.e., Hop-clique) where the central node is a passive internal adversary, and it needs to know the relative position of its one-hop neighbors. The one-hop neighbors may or may not be adversarial, but fail to detect the passive intrusion and continue to let the passive internal adversary receive routing messages.

For the central node H of a H-clique, it is not hard for H to know its neighbors' rel-

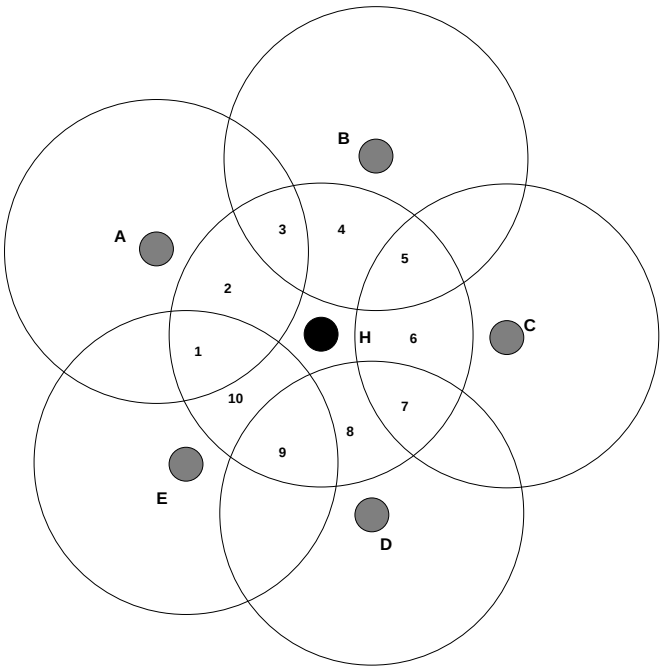

Figure 4.2: **Identifying one-hop neighbors' relative positions**

ative position (relatively clockwise or counter-clockwise). As depicted in Figure 4.2, with the help of several neighboring external adversarial nodes (e.g., sensors) H is able to know the relative positions of its one-hop neighbors in a quite precise manner. For example, if a node's transmission is overheard by H, sensor A, and sensor E, then the node is in area 1. Similarly, if a node's transmission is overheard only by H and sensor A, then the node is in area 2, and so on. More sensors and more careful configuration of the sensors will result in more precise relative position measurement.

To make the thing even worse, the passive attack is cumulative. That is, composition of H-clique attacks makes the attack more effective. As shown in Figure 4.3, a mobile node cutting through two H-cliques is detectable by the adversary if relevant routing messages are intercepted. Figure 4.4 shows that H-cliques who know their relative positions can combine multiple motion-cuts together to obtain more precise information about mobile nodes' motion patterns. Therefore, a few passive in-

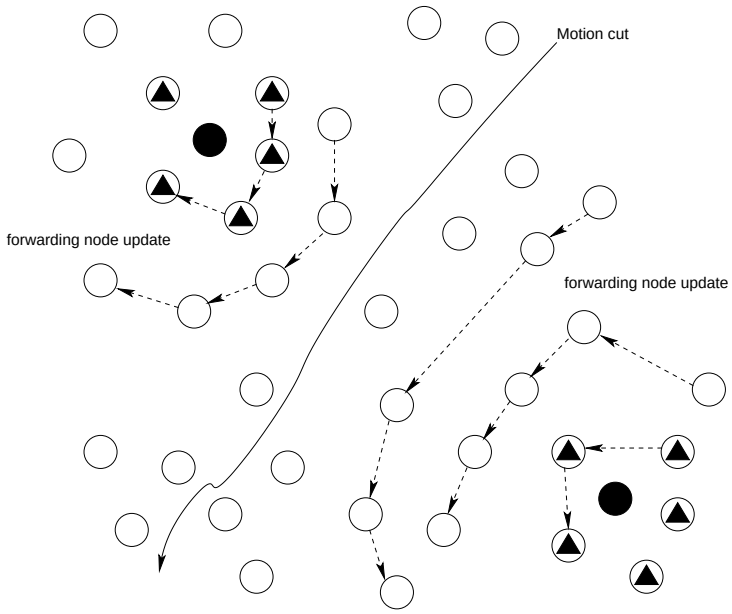

Figure 4.3: **H-clique attack: a motion cutting through two H-cliques is detectable from forwarding node updates**

ternal adversaries can effectively launch motion pattern inference attacks against the entire network. Both proactive routing schemes (e.g., distance vector and link state) and on-demand schemes (e.g., DSR and AODV) are vulnerable to such passive attacks. Combined with out-of-band information like geographic positions and battlefield boundaries, such attacks may gradually compromise motion pattern privacy of active network nodes.

4.5 Illustration through simulations

We illustrate the feasibility of the motion pattern inference attacks through simulations on AODV routing protocol [108]. Our simulation runs on the GloMoSim simulation platform [137]. GloMoSim is a packet level simulator for wireless and wired networks. The distributed coordination function (DCF) of IEEE 802.11 is used as the MAC layer in our experiments. It uses Request-To-Send (RTS) and Clear-To-Send (CTS) con-

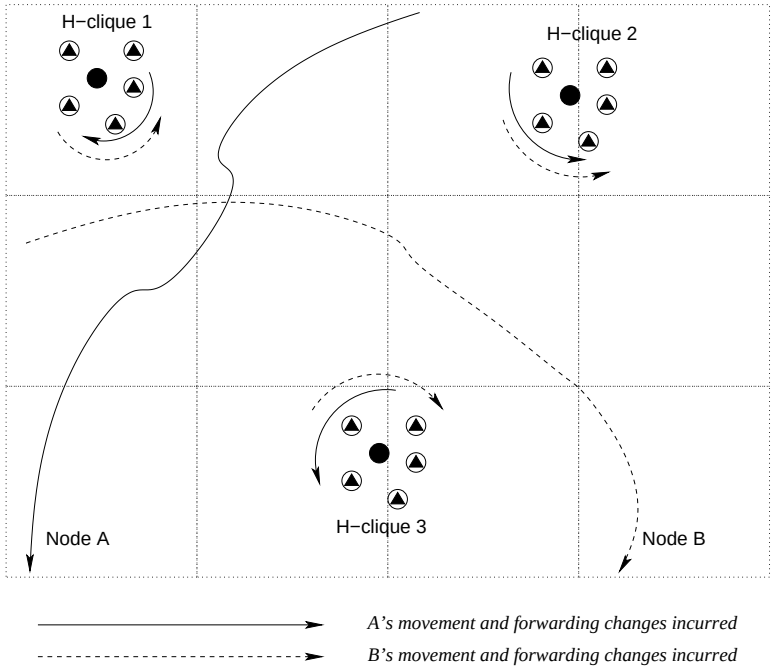

Figure 4.4: **Coalesceable H-clique attack: More H-cliques can obtain more precise motion patterns**

trol packets to provide virtual carrier sensing for unicast data packets to overcome the well-known hidden terminal problem. Each data transmission is followed by an ACK. Broadcast data packets are sent using CSMA/CA only. The radio uses free-space fading model and has characteristics similar to a commercial radio interface (e.g., Lucent's WaveLAN). The channel capacity is 2 Mbits/sec.

We simulate a scenario where a target node moves straightly across a network with many fixed nodes. While moving, the target node periodically communicates to other nodes. In the meantime, internal adversaries are presented in the network. In our simulation, we study two simple cases to illustrate the attack. In the first case, the target talks to one destination and there is only one adversary. In the second case, more adversaries exist in the networks, and the target talks to two destinations. Through the two cases, we demonstrated that with a certain number of adversaries (which are ca-

pable of communicating with each other), in a bounded time, motion pattern inference is possible.

In the simulation, we use AODV routing protocol to establish paths between the target and the destinations. AODV does not protect its routing information, thus any external adversary can also launch H-clique attack. As an on-demand protocol, AODV searches the destination when communication is needed. The search procedure starts by flooding a *Route Request (RtREQ)* message for the destination. Upon receiving a request message, the destination will reply a *Route Reply (RtREP)* message that traces the reverse path of the request back to the source, establishing a path between the source and the destination. When the path breaks, e.g., a link broken due to mobility, the source will re-issue the search procedure to build a new path between the source and the destination.

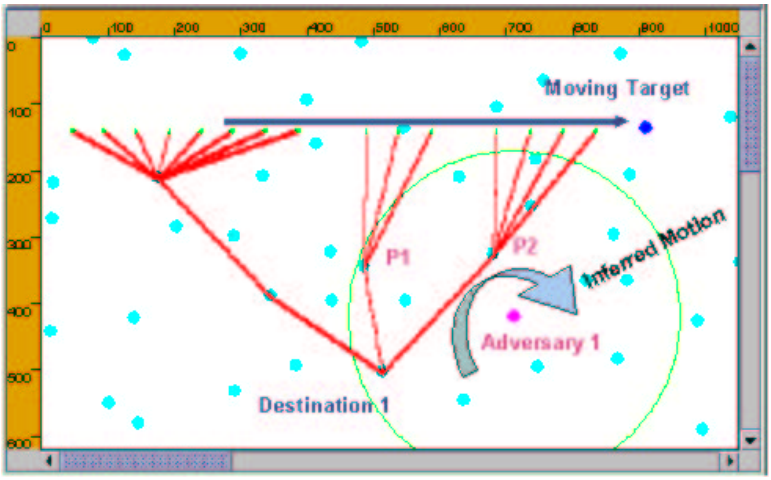

Figure 4.5: **Illustration on actual simulation animation: Motion inference with one H-clique** (Depicted nodes and ad hoc routes are from actual GloMoSim animation)

Figures 4.5 and 4.6 are snapshots of our simulations. In the two figures, the target moves from the left of the simulation field to the right. A path between the target and the destination is depicted by linked solid lines. For each communication instance, the path in use is drawn in the figure. While the target is moving, different paths are

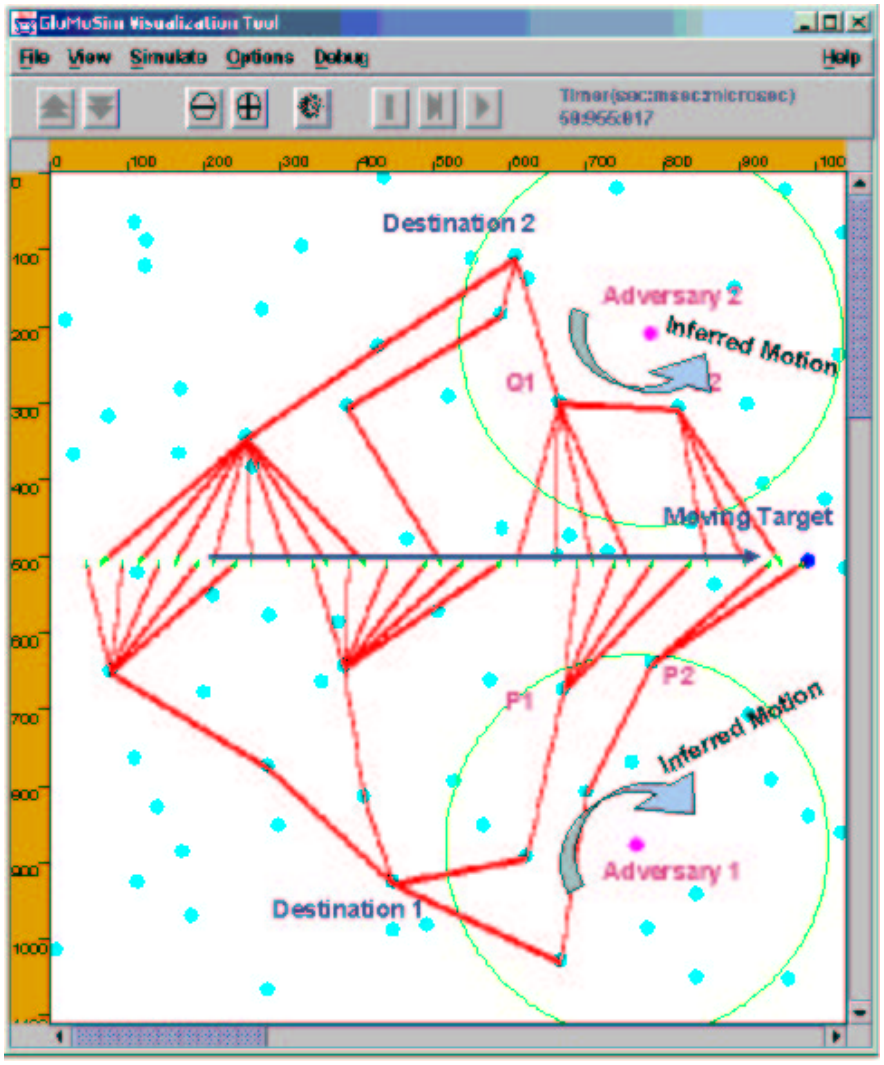

Figure 4.6: **Illustration on actual simulation animation in GloMoSim: Motion-cut attack by 2 H-cliques** (Depicted nodes and ad hoc routes are from actual GloMoSim animation)

chosen and the figure shows that the intermediate forwarding nodes have changed for several times in the simulation due to the target mobility. The adversary node and its radio range is also drawn in the figures.

In Figure 4.5, node P1 and P2 are in the radio range of the adversary. During a certain time period, the adversary node hears the path change from P1 to P2 (it can do so either through AODV path setup messages or data packets). Thus the adversary infers that there is a clockwise motion to its north-west. In Figure 4.6, the same target motion is detected by two adversaries. While the "$adversary1$" suggests a clockwise motion to its north-west, the "$adversary2$", hearing the path migration from node Q1 to node Q2, figures out that the target is moving counterclockwise to its south-west. Combining these two pieces of information, the adversaries successfully discover that there is a motion cutting through between them. If more adversaries are presented in the network, more complete and precise motion pattern will be inferred.

4.6 Summary

In summary, this chapter demonstrates that existing ad hoc routing protocols are vulnerable to mobile anonymity attacks. The work shows the necessity to devise anonymous ad hoc routing schemes to protect wireless nodes' mobile anonymity in hostile environments. In addition to traditional content privacy concerns, mobile nodes need more support to ensure their sender identity/venue anonymity, recipient identity/venue anonymity, sender-recipient identity/venue relationship anonymity, location privacy, and motion pattern privacy.

CHAPTER 5

Anonymous Data Forwarding

All animals are created equal,
but some are more equal than others.
–George Orwell "Animal Farm"

Network node identity space in a modern computer network is typically a huge anonymity set (e.g., 2^{32} IPv4 addresses, 2^{48} link layer MAC addresses, and 2^{128} IPv6 addresses). Since brute-force synchronization/broadcast over such a huge anonymity set is infeasible, we should not use node identity in order to ensure perfect identity anonymity. This chapter shows that the combination of (1) *packet stamp lookup in pairwise forwarding tables* and (2) *cryptographically strong pseudorandom generation* based on *pairwise key agreement* can be regarded as a sufficient condition of anonymous data forwarding. It is not necessary to name the network members in terms of data forwarding. The single-hop anonymous data forwarding scheme ensures perfect sender/recipient identity anonymity, perfect sender-recipient identity relationship anonymity, and strong location privacy. It is applicable to both single hop wireless networks (e.g., cellular networks and wireless LAN) and multi-hop ad hoc networks (e.g., mobile ad hoc networks and sensor networks).

The chapter is organized as follows. We first show how pairwise anonymous data forwarding is realized upon pairwise key agreement. This anonymous data forwarding

scheme also achieves strong location privacy defined in Chapter 2. Then we show this anonymous data forwarding scheme can be used to protect last-hop wireless networks.

5.1 Computationally anonymous single-hop forwarding scheme

5.1.1 Why not use current 802.11 forwarding scheme and simply encrypt entire 802.11 frame?

802.11 explicitly uses link layer addresses in packet forwarding. It is possible to implement a naive anonymous forwarding scheme by simply encrypting the entire packet/frame. We do not adopt this approach due to the following reasons.

First, we assume internal adversary and partial trust. This is in particular critical for mobile ad hoc networks, where any node must be prepared to operate in a mode that trusts no single peer. With no exception, the underlying 802.11 scheme reveals packet sender's identity (in both unicast mode and broadcast mode) to the packet receiver, and packet receiver's identity (in unicast mode) to the packet sender. Simple encryption fails to achieve sender identity anonymity and recipient identity anonymity against any internal adversary.

Second, 802.11 frame header is comprised of a number of static fields[1]. If the encryption key is not refreshed per packet, the ciphertext of these static fields is unchanged per connection and becomes a traceable static pseudonym. A traffic analyst can use the static pseudonym to correlate packets of the same connection and identify local communication patterns (e.g., number of local connections). This is a successful compromise against location privacy defined in Chapter 2.

[1] These fields are at the beginning of a packet/frame, and they are unchanged per connection. The first block (64 bits or 128 bits) is static even if we apply 3DES-CBC or AES-CBC encryption modes of operation.

5.1.2 Our scheme

Simple encryption randomizes transmitted information, but we have seen that such naive solutions fail to provide needed protection. Here we propose a computationally anonymous data forwarding scheme without using any node's identity.

In order to unlink a network member's identity and its standing location, we employ a very different approach from many existing data forwarding schemes using explicit node identity. In contrast, we give the transmission hop a *route pseudonym*. This strategy bears resemblance to virtual circuits used in Internet QoS [7]. This design achieves sender/recipient identity anonymity, sender-recipient identity relationship anonymity, and weak location privacy against external adversary. If there is an anonymous key agreement protocol, this design also achieves the same set of goals against internal adversary — neither of the two communicating principals reveals its identity to the other side.

An implementation of our scheme is comprised of three components: a pair of forwarding tables maintained by both ends of the transmission hop, and the route pseudonym. Their relationship is depicted in Figure 5.1. Instead of using explicit node identities ($\{A, B, C\}$), two communicating nodes should maintain a forwarding table with a colliding entry holding the route pseudonym. A packet sender prepends or appends its data packets with the outgoing route pseudonym in its forwarding table. A data packet is then transmitted without identifying the sender and the receiver. In a wired environment this is simply a single-hop virtual circuit. In a wireless broadcast environment things are different. All other local receiving nodes must look up the route pseudonym in their forwarding tables. The node discards the packet if no match is returned from its forwarding table.

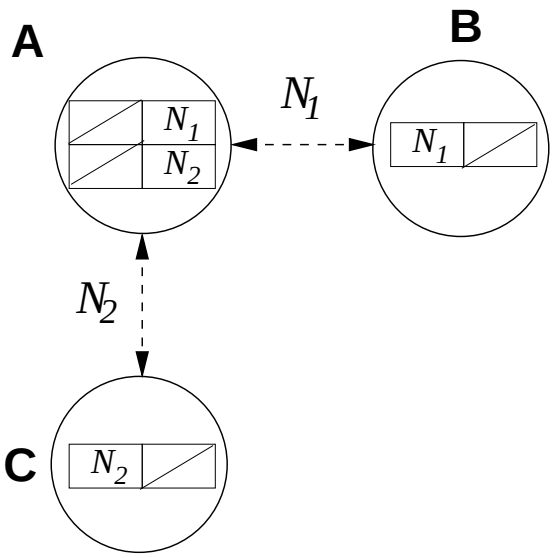

Figure 5.1: **Different approaches in packet forwarding** (Using node identity $\{A, B, C\}$ vs. using route pseudonym $\{N_1, N_2\}$)

5.1.3 Route pseudonym collision

In a wireless broadcast environment, route pseudonyms used by different connections in the same locality may collide with each other. In the ideal case, no route pseudonym collision is allowed within any forwarder's single hop neighborhood. Here we study how to enforce the constraint.

Suppose route pseudonym is selected randomly. As the chance of collision p_c decreases exponentially when pseudonym length l increases linearly (currently we select the route pseudonym length $l = 128$ bits), random selection following uniform distribution inside the pseudonym space is computationally collision resistant. For arbitrarily k randomly selected pseudonyms within a one-hop wireless transmission range, the chance of collision p_c is only

$$p_c = 1 - \frac{\prod_{i=0}^{k-1}(2^l - i)}{(2^l)^k} = 1 - \prod_{i=1}^{k-1}\left(1 - \frac{i}{2^l}\right)$$

$$< 1 - \left(1 - \frac{k-1}{2^l}\right)^{k-1} = \frac{2^{l(k-1)} - (2^l - k + 1)^{k-1}}{2^{l(k-1)}}.$$

If we expand "$-(2^l - k + 1)^{k-1}$", the first item "$-2^{l(k-1)}$" is canceled by "$2^{l(k-1)}$" in the numerator. All other items are less than the order "$2^{l(k-2)}$", and there are polynomially many such expanded items. The entire quantity is then of the order $\frac{P(\cdot)}{2^l}$ where $P(\cdot)$ denotes a polynomial. This quantity decreases exponentially as l increases linearly. In other words, each route pseudonym is computationally unique when l is large enough.

5.1.4 Hiding communication patterns: Route pseudonym update using CSPRG

If we keep using static route pseudonyms, a nearby traffic analyst can know critical information about the locality, for example, it can count the number of local connections. This is a successful location privacy attack that reveals local nodes' communication patterns. We should address such privacy problems and achieve the following effect against a $\mathcal{BPP}$ adversary:

> *Intuitively, given arbitrarily x (e.g., $x = 100$) locally intercepted data packets, the $\mathcal{BPP}$ adversary cannot see the relation among these 100 packets. That is, all possible cases are evenly distributed. The two extreme cases are: (1) all 100 packets were transmitted from one sender to one recipient, and (2) the 100 packets were transmitted from 100 distinct senders to 100 distinct recipients. These two extreme cases and all in-between cases are equally likely to the $\mathcal{BPP}$ adversary.*
>
> *In the ideal case, packets of two different types are independently transmitted; any two packets of the same type are of identical size, and they look like two random independent transmission events with random packet contents.*

The data payload field can simply be encrypted by a semantically secure encryption scheme so that the encryption result looks random to a $\mathcal{BPP}$ adversary. In addition, this security requirement demands no use of *static pseudonyms*. To foil the attack, we randomize route pseudonyms by self-synchronized route pseudonym update. Consider two nodes sharing a route pseudonym in their forwarding tables. One is an outgoing entry, and the other is an incoming entry. As long as these two entries are appropriately synchronized, the pseudonym can be constantly changed to other pseudorandom but locally unique values.

Route pseudonym update explores the concept of *unpredictability in polynomial time*. This concept means that no Turing-complete algorithm is able to differentiate a cryptographically strong pseudorandom ensemble from a truly random ensemble in polynomial time. The pioneer work done by Yao[145], Blum, and Micali[19] illustrates the relation between one-way functions and pseudorandom number generators. They showed that cryptographically strong pseudorandom bit generators realized on top of one-way functions can pass next-bit-test. Thus any polynomial time statistical test achievable by a $\mathcal{BPP}$ adversary cannot distinguish the next pseudorandomly generated bit from a truly random bit.

Slow but provably secure pseudorandom bit sequences can be constructed using hardcore predicates [55] of a one-way function. In particular, as the hardcore predicate for any one-way function have been discovered, cryptographically strong pseudorandom generators are constructible from any one-way function [61]. However, due to performance concerns, we can use keyed fast one-way functions f (e.g., HMAC, 3DES, AES) to generate pseudorandom block sequences instead of bit sequences.

Now we require the shared pseudonym (e.g., N_1 in Figure 5.1) to be a shared secret seed. This requires an anonymous key agreement protocol (will be presented in Chapter 6) between two one-hop neighbors. Route pseudonym sequence is generated

by feeding the shared secret seed N_1 into the fast one-way function f, then feeding the output back to the input repetitively. In other words, the i-th pseudonym is

$$n_i = \underbrace{f(f(\cdots f}_{i}(N_1)\cdots)) = f^i(N_1)$$

where f is a keyed well-known fast one-way function, e.g., $f(n_i) \triangleq AES(N_1, n_i)$ which also uses the secret seed N_1 as the key. The unpredictability of such fast pseudorandom sequence is proven by Shamir [131], but in a less-formal sense than computational complexity. Another fast practical pseudorandom sequence generator can be found in X9.17 [4] (See Appendix A.1).

The two ends of a hop should update the shared route pseudonym per forwarding packet for a reliable transmission. For a unreliable transmission, at least two candidate schemes are useful: (1) If tight time synchronization is feasible, that is, difference between the two system clocks is smaller than the delay to transmit the smallest packet on their network interface, then both ends can agree to update the route pseudonym per short interval t_{int}; (2) The sender stamps a non-decreasing sequence number seq on each packet payload. The receiver computes $n_{seq} = f^{seq}(N_1)$ based on current pseudonym. The values for seq are not necessarily consecutive [77]. If the difference between two consecutively received sequence numbers is reasonably small, the incurred computational overhead is acceptable.

In a nutshell, with appropriate route pseudonym update and payload encryption, any two data packets are bit-strings not differentiable from truly random bits by a $\mathcal{BPP}$ adversary. This means the scheme achieves *strong location privacy* against a $\mathcal{BPP}$ adversary (if all data packets are of the same size).

5.2 Computationally secure single-hop forwarding scheme with asymmetric anonymity support

If two communicating parties are asymmetric, for example, one is a mobile client and the other is a fixed server in the wired infrastructure. We can devise an anonymous single-hop forwarding scheme based on client-server model. In the scheme the anonymity support is asymmetric: while the anonymity for any mobile client is ensured, the server's anonymity is not protected. This scheme is suitable in single-hop wireless environment (e.g., wireless LAN, cellular network) as the server in the infrastructure may not need mobile anonymity supports.

The packet forwarding scheme is identical to the one described in Section 5.1, except the key establishment procedure explicitly authenticates the server, but hides the mobile client's identity. Assuming a PKI-conforming environment, the client uses the server's certified public key to encrypt a random secret value, then the server can use its own private key to see the value by decryption. Existing standards like TLS [37] and WTLS [142] already define the details of such server authentication and key exchange schemes. In particular, we refer to WTLS' class 1 authentication and key exchange protocol or any equivalent scheme. Such a scheme will establish a shared secret seed used in our anonymous data forwarding scheme.

Example 10 ***(Need of anonymity protection in wireless LAN)*** *Most 802.11 (Wi-Fi) access points have a built-in feature called* MAC Address Filtering*, which is widely used and allows the network administrator to enter a list of MAC (Media Access Control) addresses that are allowed to communicate on the network.*

On the other hand, most 802.11 (Wi-Fi) NICs allow the user to configure the MAC address of the NIC in software. Therefore, if an adversary can sniff the MAC address

of an existing network node, it is possible to join the network using the MAC address of the victim node. This shows anonymity protection is needed in wireless LAN as well.

The anonymous data forwarding scheme proposed here can be used as a simple solution. A mobile client and the access point can hide their identities by using 802.11 broadcast address (all 1's) and implement the route pseudonym stamp and lookup services proposed in this chapter.

5.3 Summary

This chapter describes a practical anonymous data forwarding scheme which is also computationally secure. The scheme does not use node identity[2], thus achieves *perfect sender identity anonymity*, *perfect recipient identity anonymity* and *perfect sender-recipient identity relationship anonymity*, even against the other principal (assuming anonymous key agreement). The scheme also implements route pseudonym update to achieve *strong location privacy*. It does not use any static pseudonym, which indeed reveals communication patterns as if a real identity is used.

[2] ***Throughout this dissertation, perfect sender identity anonymity, perfect recipient identity anonymity, and perfect sender-recipient identity relationship anonymity are accomplished because we never explicitly use mobile node's identity in anonymous data forwarding and routing***. This is because intercepting any transmission event does not decrease the uncertainty entropy of sender/recipient's identity. From now on this claim is assumed.

CHAPTER 6

Anonymous Routing Resilient to Passive Attacks

Distinct things are likely to
be identified and destroyed.

–Chinese proverb

Anonymous data forwarding described in Chapter 5 only addresses single-hop anonymous communication. In a multi-hop wireless network, e.g., a mobile ad hoc network, we need to devise new routing schemes to ensure anonymity guarantees for the network.

This chapter presents an ideal anonymous communication model (TIMBA) and a follow-up practical routing scheme (ANODR) against a passive adversary that follows our protocols, but tries to compromise mobile anonymity (i.e., "honest-but-curious" adversary). Like Chapter 5, we observe that network node identity space in a modern computer network is typically a huge anonymity set (e.g., 2^{32} IPv4 addresses, 2^{48} link layer MAC addresses, and 2^{128} IPv6 addresses). Since brute-force synchronization/broadcast over such a huge anonymity set is infeasible, we should not use node identity in order to ensure perfect identity anonymity. We show that a combination of network and cryptographic mechanisms, namely the combination of (1) *on-demand approach*, (2) *virtual circuit*, (3) *pairwise key agreement only between sender and recipient, and between two neighboring forwarders*, and (4) *boomerang onion*, can be regarded as a sufficient condition of anonymous routing. It is not necessary to name

the network members in terms of routing[1], yet routing is feasible in an on-demand approach using global and local trapdoors.

The chapter is organized as follows. (1) We first describe two anonymity models, namely *Time Interval and Broadcast Anonymity (TIBA)* model and *Time Interval and Multi-hop Broadcast Anonymity (TIMBA)* model, to ensure perfect mobile anonymity. (2) Then we present ANODR as the practical routing scheme. It is a compromise between the ideal models and the real world.

6.1 TIBA & TIMBA: towards scalable multi-hop anonymity model

The notations used in this work is shown in Table 6.1. The concept of "scalability" refers to how a solution pays smaller cost to solve a problem when the size of the problem increases. However, "scalability" is not *formally* defined in many literatures. A solution is clearly scalable if it pays constant cost even when the size of the problem increases. However, such solutions are normally unrealistic. In this dissertation, our scalability goal is like this: By "the time complexity of solution A is scalable to problem B", we mean the time cost of solution A increases logarithmically/linearly when the size of problem B increases linearly/exponentially. Similarly, we can define scalability in terms of storage space complexity and communication complexity.

In this section we will present a scalable anonymity model using a stepwise approach. A series of models are described in a way that they are progressively towards more scalable to and more consistent with our network model. The key idea of our design is hiding real end-to-end connection events in a *uniformly distributed transmis-*

[1] So far to our knowledge, ANODR is the first routing scheme that does not use node identity in routing at all. Even in existing schemes using virtual circuits (e.g., ATM [7]), multi-hop routing explicitly identifies the previous stop and next stop's identities (e.g., in ATM's signaling phase). Therefore, any internal adversary can compromise anonymity and location privacy in its neighborhood. TIMBA/ANODR is not vulnerable to such attacks.

Table 6.1: **Table of variables and notation**

$\langle pk_A, sk_A \rangle$	Node A's public/private key pair
k_A	A key only known by node A
k_{AB}	A symmetric key shared by node A and B
$y = f(x)$	A one-way function f outputs y on input x
$y = f^{-1}(pk, x), \;\; x = f(sk, y)$	Trapdoor one-way function f in an asymmetric cryptosystem (the encryption/verification function f^{-1} uses pk, while the decryption/signing function f uses sk)
$y = \{x\}_{pk}, \;\; x = [y]_{sk}$	Alternative notion of $y = f^{-1}(pk, x)$ and $x = f(sk, y)$ (if f and f^{-1} are well-known)
$y = f(k, x), \;\; x = f(k, y)$	Trapdoor one-way function f in a symmetric cryptosystem (the same one-way function f is used in both encryption and decryption on the same trapdoor key k)
$y = k(x), \;\; x = k(y)$	Alternative notion of $y = f(k, x)$ and $x = f(k, y)$ (if f is well-known)
N_A, N_A^i	Nonce or nonces chosen by node A
h_D	Network diameter — maximal distance between any two nodes (distance is the minimal hop count between the two nodes)
RtREQ	Route Discovery Request Packet
RtREP	Route Discovery Reply Packet
RtERR	Route Maintenance Error Packet
src	A well-known string (e.g., "i'm src") notifying the source
$dest$	A well-known string (e.g., "u r dest") notifying the destination

sion event space. It is implemented along two dimensions: (1) At spatial dimension, each receiver is equally likely to be the real receiver of an end-to-end connection event when the sender broadcasts to all network members[2]; (2) At temporal dimension, distinct transmission events are implemented as evenly distributed staccato transmissions along the timeline. A node must send out a pre-defined number of packets per time interval T_{int}[3].

From now on we will call this two policies as (1) *broadcast policy* and (2) *time*

[2] In a multi-hop network, we will use hypercube and efficient control flow to address scalability issues.

[3] A more general design is to let all network members share a common seed, thus share a pseudo-random sequence of bits. Each bit corresponds to a time interval. All network members must maintain radio silence if the current bit is 0, or must broadcast if the current bit is 1. An all-1 sequence is actually the complement of an linear congruential generator "$ax + b$ mod 1".

interval policy. For practical reasons, we will focus on the broadcast policy in our practical protocol design of ANODR. The time interval policy is mainly proposed to address timing analysis (the typical causality correlation attack). It is not needed if the adversary is not capable of doing timing analysis. Although in practice it incurs unreasonable communication overheads, in theory it gives the upperbound of anonymity protection against timing analysis. This upperbound has not been identified in all previous MIX research efforts.

6.1.1 TIBA: a one-hop ideal model for reference purpose only

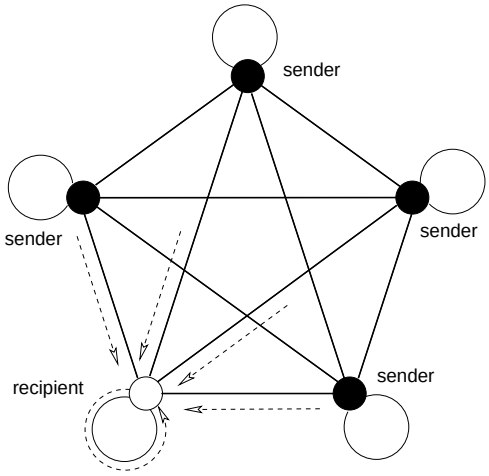

Figure 6.1: **TIBA: sender anonymity**

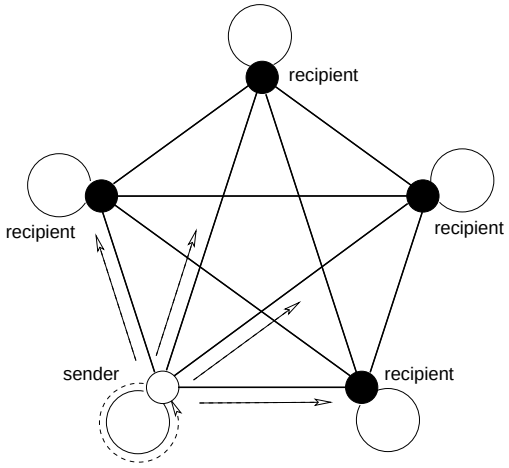

Figure 6.2: **TIBA: recipient anonymity**

Here we present *Time Interval and Broadcast Anonymity* (TIBA) model. This model is not practical, but it illustrates our design philosophy–"perfect anonymity can be achieved by hiding real transmission events in a uniformly distributed transmission event space". In TIBA, we realize the "timing interval policy" to address timing

analysis. Combined with a dummy traffic design protected by Vernam Cipher, TIBA ensures perfect mobile anonymity in a fully-connected network.

Key management assumption: TIBA assumes pairwise key agreement in the network. This requirement, not a requirement of public key cryptosystem or equivalence, is the base of our anonymity design. However, in practice, pairwise key agreement must be realized by a cryptographic method like public key cryptography. For example, we can build an offline certification authority who signs certificates for network members. Each network member chooses its own personal public/private key pair, but must have its personal public key certified before joining the network. Once issued, a node's certificate is public. Any certificate can be cached on any node, or can be circulated in the network. It is important to note that such public key knowledge is *a priori* before TIBA is applicable. Knowing the public key does not imply anything about anonymous data forwarding or routing. Like knowing the public ID space $\mathcal{I}$, knowing *a priori* public key of any network member does not affect mobile anonymity. This work seeks to minimize the difference between *a posteriori* and *a priori* knowledge, rather than to eliminate *a priori* knowledge. Network members can be loaded with *a priori* materials if such *a priori* materials does not compromise mobile anonymity.

Single-hop data flow: Traffic analysis is addressed by a fully connected network with uniformly distributed transmission events. (1) At temporal dimension, network lifetime is divided into many short intervals of size T_{int}. During each interval, each node must send/receive d packets, some of them are dummy packets (dummy flag=1) if there is no real transmission (dummy flag=0). Hence out-/in-bound transmissions of every node are uniformly distributed over time. (2) At spatial dimension, every real/dummy transmission is broadcast to all nodes (including itself) in the same time interval. Broadcast in such a network simply means that when a node sends a

message, all nodes will receive a copy. In a wired media, broadcast can be realized through (unordered) multiple unicasts, while in an open space wireless media, a single omnidirectional transmission covers all nodes. During each interval, the order of transmissions from a sender to different recipients (or from different senders to a recipient) is insignificant. This way, out-/in-bound transmissions of every node are uniformly distributed over space.

Case I: A message transmitted in wired TIBA network is prefixed with a dummy flag. A sample conceptual packet is given below:

← Vernam Cipher using agreed key →	
dummy flag (1 bit)	message (arbitrarily long)

The dummy flag field is enciphered with Vernam Cipher. In practice, the one-time pad used in Vernam Cipher can be replaced by a cryptographically strong pseudorandom ensemble generated from the agreed key.

Case II: A message transmitted in wireless TIBA network covered by one-hop omnidirectional radio is prefixed with a route pseudonym. Dummy flag is not needed here because the dummy condition is implies by an empty forwarding table lookup in the anonymous data forwarding scheme (Chapter 5). This can be implemented by embedding a truly random route pseudonym that is out of synchronization of any route pseudonym sequences in use. A sample conceptual packet is given below:

← cryptographically strong pseudorandom ensemble →	
route pseudonym (fixed length)	message (arbitrarily long)

Fig. 6.1 and 6.2 depict the per-interval per-node case of TIBA for a fully-connected network comprised of $N = 5$ nodes. At the sender's side, it sets the dummy flag or

the route pseudonym field according to application's demands. Then only the real recipient can see a packet with (dummy flag=0) or find a matching route pseudonym from its forwarding table. It is important to note that we are not discussing message privacy here. *All messages transmitted in TIBA can be plaintexts without encryption*— we are only interested in the property that real sender/recipient cannot be identified by adversaries via global traffic analysis. Message privacy is not the subject studied in this work.

Analysis: If we compare Figure 6.1 and 6.2 with Figure 2.4, we can find an intuitive but straight-forward proof that TIBA ensures perfect mobile anonymity against a $\mathcal{BPP}$ adversary.

Even though the $\mathcal{BPP}$ adversary is also capable of timing analysis, he is stopped by the time interval policy. As a $\mathcal{BPP}$ adversary cannot distinguish cryptographically strong pseudorandom ensembles in the real traffic from truly random ensembles in the dummy traffic, any transmission event looks like a random event. For a $\mathcal{BPP}$ timing analyst, a transmission event happens simply because of the time interval policy — he knows the rule *a priori*, thus his gains no *a posteriori* knowledge after the transmission event happens.

Cost complexity: In TIBA, a message of l packets long is delivered within $\lceil \frac{l}{d} \rceil \cdot T_{int}$ time. Given a fully-connected network with N nodes, each node must send/receive $d \cdot N$ messages per T_{int}.

6.1.2 α-TIMBA: a multi-hop ideal model based on hypercube

TIBA model requires a fully-connected network, which is obviously not scalable. By using *local broadcasts* and *hypercube*, we translate it into a scalable model, namely *Time Interval and Multi-hop Broadcast Anonymity* (TIMBA) model. By "local broad-

cast", we mean broadcasting to all the directly connected nodes (one-hop neighbors). The network is no longer a fully connected one due to multi-hopping, but can be better interpreted using a hypercube structure. In a TIMBA network we assume a $\mathcal{BPP}$ traffic analyst is right beside each legitimate node, and they collaborate globally. In particular in a wireless TIMBA network a $\mathcal{BPP}$ traffic analyst and its one-hop adversarial neighbors form a traffic analysis "cell" corresponding to a vertex in the underlying graph G. Thus the hypercube structure information (e.g., vertex labels) is considered *a priori* topological information that is publicly known.

In this section we present the α-TIMBA model for anonymous uni-directional traffic in a static hypercube network. Readers with background knowledge about Chaumian MIX-net [28] may simply treat this α-TIMBA model as a tutorial model similar to Chaum's design, which is suitable to deliver limited amount of messages in a fixed network. However, it does not address three important demands: (1) network topology changes due to mobility; (2) bi-directional packet flows; and (3) real time packet flows with large quantity of data, e.g., multimedia streams. In the next section we will present the standard TIMBA model to address network dynamics and bi-directional real-time data traffic.

Definition 4 ***(Hypercube)*** *Let Q_r be a hypercube of r-dimension. Q_r is inductively constructed by the following algorithm:*

- *Q_0 is a single point with 2^0 vertex.*
- *Q_1 is a line segment with 2^1 vertexes.*
- *Given Q_{r-1} with 2^{r-1} vertexes, we label each vertex of Q_{r-1} with $(r-1)$-bit binary string. Then Q_{r-1} is duplicated as Q'_{r-1}. In the original Q_{r-1}, each vertex label is prefixed with 0. In Q'_{r-1}, each vertex label is prefixed with 1. Q_r is constructed from Q_{r-1} and Q'_{r-1} by connecting them: A vertex in Q_{r-1}*

is connected with a vertex in Q'_{r-1} if and only if their labels differ at the most significant bit. □

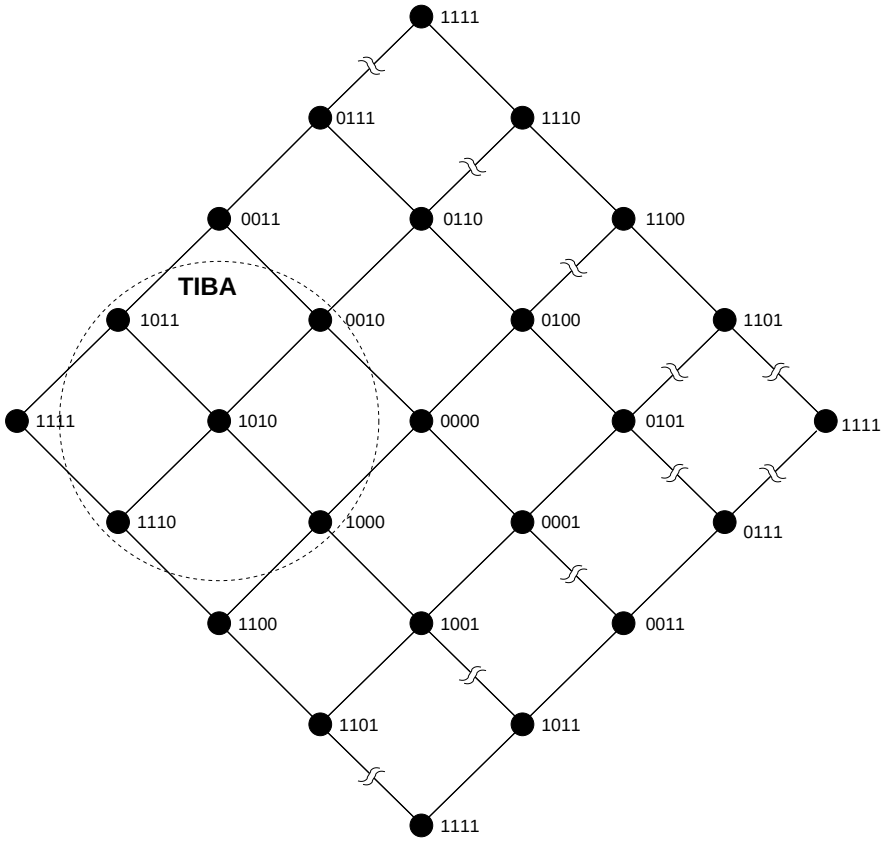

Figure 6.3: **TIMBA: A planar projection of Q_4**

In Q_r, there is an edge between two vertexes if and only if their labels differ in exactly 1 bit. The dimension r is both the degree of each vertex and the diameter h_D of the entire network. As an example, Figures 6.3 shows Q_4 in a planar graph. If we treat a node and its one-hop neighbors as a *local network*, then the one-hop broadcast mechanism of the TIBA model is applicable to this local network where locally $N_{loc} = r + 1$. That is, in α-TIMBA each node only sends d data packets to each

of its $r = \log_2 N$ neighbors per time interval T_{int}. This complexity decrement satisfies the scalability requirement.

Key management pre-requisite: Like Chaumian MIX-Net, an α-TIMBA sender (source) knows an encryption key of every intermediate forwarder and the final recipient. To address content correlation, any TIMBA packet transmitted on two consecutive hops must be changed by decryption and of a uniform fixed size.

Multi-hop data flow: In a multi-hop network, any packet forwarded on consecutive hops must be (re-)encrypted or (re-)decrypted to stop content correlation attack. This implies message payload must be of uniform size and must be (re-)encrypted per hop even though message privacy is not the subject studied in this work. This is a major difference between one-hop models like TIBA and multi-hop models like TIMBA.

In a static hypercube network, a sender can randomly select an arbitrary path towards its recipient, then use Chaum's design[4] to deliver the message [28]. In other words, when every node has a well-known public key, and if a sender s needs to send a packet m to recipient d via a sequence of static nodes $\{q_{n+1}, q_n, \cdots, q_2, q_1\}$, then the sender forms a message

$$\{q_n, N_s^n, \{\cdots \underline{\{q_1, N_s^2, \{d, N_s^1, \{m, N_s^0\}_{pk_d}\}_{pk_{q_1}}\}_{pk_{q_2}}\} \cdots\}_{pk_{q_{n+1}}}$$

Such layered cryptographic data structure is called "*onion*" in this work. Each forwarder strips off one layer of the cryptographic onion and forwards the stripped result to next hop. By this sender-centric method, a sender knows the entire path in a

[4] In Chaum's original proposal [28], the size of the ciphertext sent from the sender is proportional to the number of intermediate nodes. Park et al. [105] constructed a scheme based on El Gamal encryption, where the ciphertext is always just two El Gamal blocks long. The encryption workload and resulting ciphertext length were independent of the number of intermediate nodes.

fixed network topology, but a forwarder and the recipient's knowledge about the traffic flow is localized within one-hop.

Analysis: α-TIMBA is merely a tutorial model. In α-TIMBA, each network sender must know the entire network topology *a priori*, otherwise the sender cannot find a pre-computed route to reach the destination if network members are mobile. Because of this *a priori* network topology knowledge stored on each sender, all mobile anonymity protection is vulnerable to single point of compromise of any sender. For this reason, we will give formal cryptanalysis for the followed standard TIMBA model instead of the tutorial α-TIMBA model. Here we only give an intuitive explanation of α-TIMBA's communication behavior.

At each hop of data forwarding, α-TIMBA's uniform traffic pattern inherited from TIBA effectively hides real transmission events. On the next hop along a real multi-hop path, the same real packet can wait for the next interval, or the earliest chance to be forwarded if congestion happens. The uniform traffic pattern stops timing analysis.

In each locality, a $\mathcal{BPP}$ adversary is unable to correlate packet contents forwarded on consecutive hops if it cannot break a semantically secure encryption[5]. To thwart correlation on message payload size, all payloads must be padded or fragmented to a predefined uniform size. This stops content correlation attacks.

Cost complexity: Given a hypercube network with N nodes, each α-TIMBA node only sends/receives

$$d{\cdot}r = d{\cdot}log_2 N$$

messages per T_{int}. On the other hand, the network diameter $h_D = r = log_2 N$ is the maximal distance between any two nodes. A message of l packets long will be

[5] Raw RSA is not semantically secure. Pfitzmann and Pfitzmann [114] fixed a weakness in Chaum's original scheme.

delivered from a sender to its recipient within $O(\log N)$ bounded latency

$$T_{int} \cdot \left\lceil \frac{l}{d} \right\rceil \cdot h_D = T_{int} \cdot \left\lceil \frac{l}{d} \right\rceil \cdot log_2 N.$$

6.1.3 Standard TIMBA for bi-directional traffic in dynamic networks

α-TIMBA is not applicable to a dynamic network studied in this work. First, as we described in Chapter 2, mobility will randomly re-organize network nodes in the underlying network graph. Then message delivery fails because the re-organized forwarding nodes may not be the nodes selected by the source/sender. Second, the destination/recipient cannot send back unicast reply because it does not know who is the source/sender. Third, the incurred computation overheads are too high to secure real-time data traffic. These challenges are answered by the standard TIMBA model.

As shown in years of ad hoc routing research, the on-demand approach effectively addresses mobility and network dynamics if mobility speed is within the limit that characterized by round-trip communication latency in the network. An on-demand routing scheme typically consists of query and reply phases. In the query phase, the source of a communication, if it does not know a path to the destination, floods a request message to the network. This request message will be forwarded by each node in the network when the node sees the message for the first time. When the destination receives such a request message, it will start the reply phase by sending a reply message to the source. The reply message traces back the path the request comes from. The reply message also sets up the path for subsequent data transmissions. If route outage happens, the source will periodically initiate such on-demand route discovery procedure to find a valid path. On the other hand, a route error report scheme is used in on-demand routing to notify the source about route outage at real time.

A significant distinction of standard TIMBA is that we do not use Chaum's sender-

centric design. Instead, the layered cryptographic onion is assembled by distributed means based on the on-demand query-reply design. This design is feasible upon introducing control flows. During each interval, each node constantly floods the entire network with real and dummy control packets in equal probability. The three types of control packets are request (REQ), reply (REP), and error report (ERR). Without loss of generality, let's say each node must originate 1 REQ packet, 1 REP packet, and 1 ERR packet per interval T_{int}. Whether such a control packet is real is determined by the unpredictable application demand (a *de facto* coin-flip). The packet, whether real or dummy, will travel the hypercube and reach all nodes.

Soft-state on-demand design: Unlike α-TIMBA which is a stateless design, standard TIMBA is a soft-state design where each node must maintain a record for each connection. All records are recycled upon a system parameter T_{state}. Figure 6.4 illustrates the state information maintained on consecutive hops along an on demand connection. As described below, standard TIMBA uses dedicated REQ/REP control flow to establish anonymous virtual circuits and per-hop encryption keys in an on-demand manner.

Each connection is identified by a unique sequence number $seq\#$ at each node en route. The soft-state status maintained on each node is of the following format:

$$\langle seq\#, onion_{old}, personal\_trapdoor\_key, onion_{new}, \\ public_{upstream}, key_{forUpstream}, private_{me}, key_{downstream} \rangle$$

Each field of the soft-state status will be explained one-by-one below. In particular, $seq\#$ is explained in "global trapdoor". The triple $\langle onion_{old}, personal\_trapdoor\_key, onion_{new} \rangle$ is explained in "multi-hop REQ control flow". The quadruple $\langle public_{upstream}, key_{forUpstream}, private_{me}, key_{downstream} \rangle$ is also explained in "multi-hop REQ control flow".

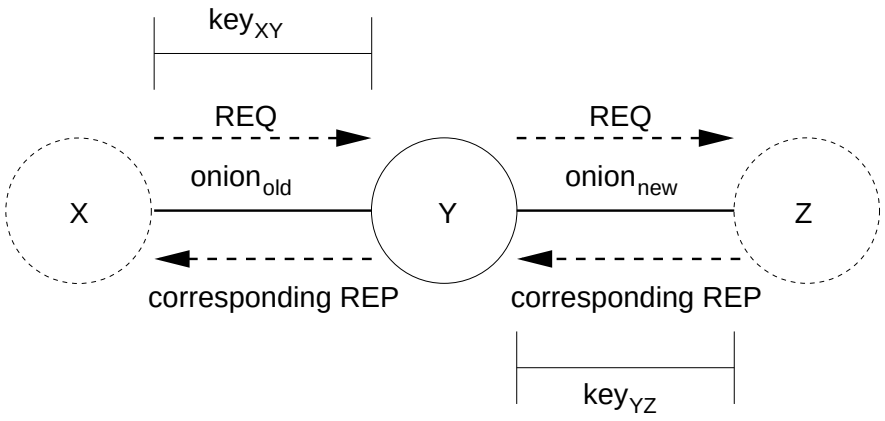

Figure 6.4: **Standard TIMBA: soft state design**

Key management pre-requisite: Like α-TIMBA, key agreement can be achieved by public key cryptography. Two sets of keys are needed:

- To address content correlation, each node needs to share a per-link encryption key *only* with its upstream node and downstream node per route discovery. To defend passive external adversary, there is no need to share keys with other neighbors and remote nodes.
- To anonymously notify the real recipient via global trapdoor (described right below), each node needs to share an encryption key with each of its real recipient to communicate with. A recipient accepts a packet only if the decrypted message shows $dest$.

Compared to MIX-Net, this key management requirement is simple. Only sender/source and recipient/destination need *a priori* keying materials. The per-hop encryption key is established in the on-demand route discovery process.

Global trapdoor and anonymous key agreement: Because a peer node

can now either be a recipient or merely a forwarder, a global trapdoor is used to notify the real recipient about its role. The plaintext is a concatenation of a well-known $dest$ string (e.g., "you are the destination!") and a random nonce $seqnum$. The random nonce $seqnum$ ensures that the global trapdoor can be used as a (computationally) unique sequence number. This random nonce also ensures that multiple REQs towards the same destination cannot be correlated by a $\mathcal{BPP}$ adversary. For the first time contact, a specific part of the random nonce (say, the first 128-bit) can be used as the symmetric key agreement.

The plaintext is then encrypted by the destination's public key if this is the first-time contact, or by the agreed key after first-time contact. Here we require trapdoor one-way permutations that permute the plaintext nonce into ciphertext nonce. Then only the intended recipient can decrypt this ciphertext and see its role. If the sender embeds random strings other than $dest$, then the global trapdoor is a dummy that will be ignored by all network members.

Multi-hop REQ control flow: Figure 6.5 depicts the control flows in Q_4. For the ease of presentation, only two possible paths are depicted between source venue/vertex #1010 and destination venue/vertex #0110. The entire procedure is described below.

The sender assembles an REQ control packet per interval, in a format given below:

	encrypted with ← recipient's key →	set local trapdoor ← per-hop →	for per-hop ← key agreement →
REQ (2 bits)	global trapdoor (fixed length)	onion (fixed length)	one-time public key (fixed length)

If this REQ is real, the global trapdoor can be opened by the intended recipient. Otherwise, the REQ is dummy, that is, the global trapdoor is randomly encrypted so that nobody can decrypt it into the $dest$ string (with non-negligible probability).

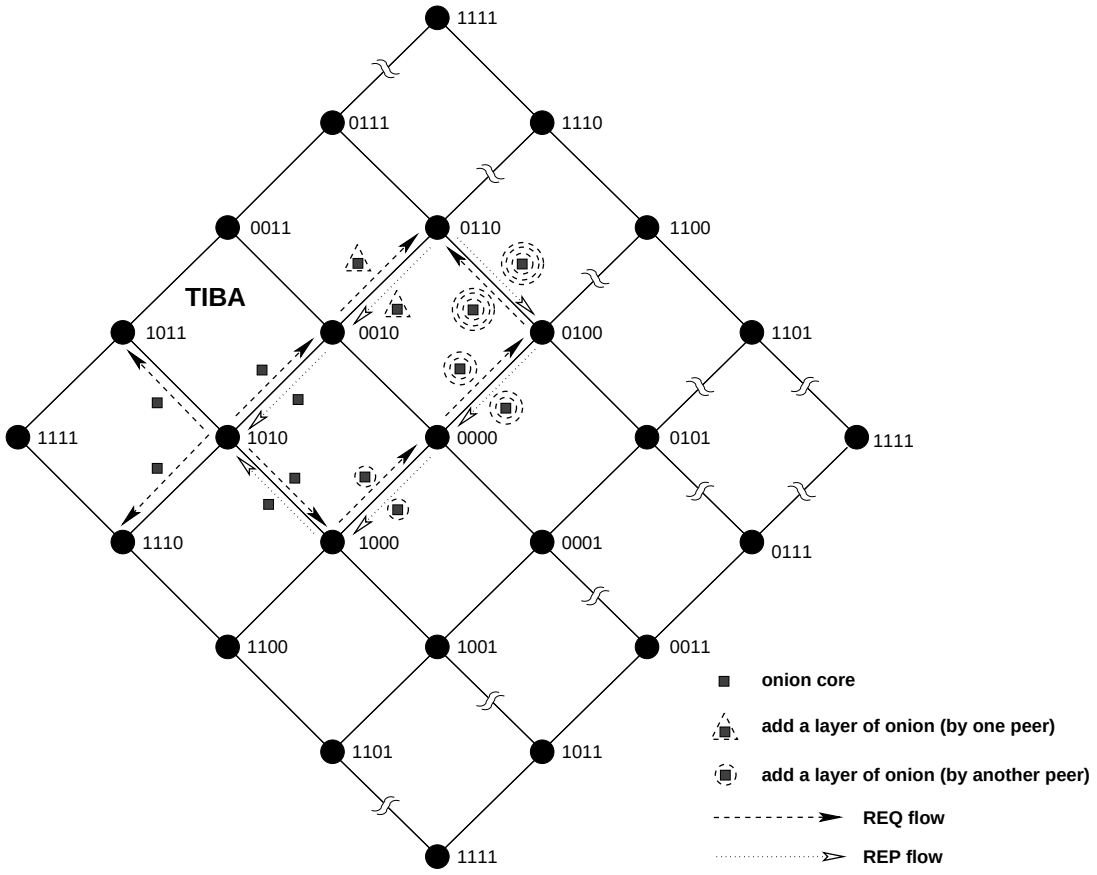

Figure 6.5: **Control flows in standard TIMBA** (The increased onion size is depicted for intuition. In standard TIMBA the onion size is fixed)

The real recipient accepts an REQ if its decryption shows $dest$. Nevertheless, it must forward the REQ packet just like other nodes.

The global trapdoor is also used as a (computationally) unique sequence number. Soft-state information is stored under this unique sequence number on every node en route. Besides, this sequence number can avoid "broadcast storm" problem [99]. Without this sequence number, an REQ packet may circulate in the hypercube network forever. The REQ forwarding procedure is loop-free if the network adopts the following strategy: each node forwards an REQ packet with unique sequence number if and only if it sees the sequence number for the first time[6]. In other words, at each stop, REQ packets with previously seen sequence numbers are suppressed. Given a unique sequence number, at each stop the forwarder receives up to r real REQ packets from its neighbors. It randomly chooses one of them for further forwarding, embeds its own trapdoor in the onion of the selected REQ, then locally broadcasts the modified REQ. This forwarding strategy ensures that an REQ packet with a unique sequence number is forwarded and only forwarded once on each node.

For the onion, the original sender can embed unique random bits in the onion-core as long as it remembers what it has sent. At each stop, the forwarder uses a well-known trapdoor one-way permutation family to embed one more secret trapdoor in the onion.

$$onion_{new} \longleftarrow f(personal\_trapdoor\_key, onion_{old})$$

[6] By inspecting the global trapdoor field, a global timing analyst can differentiate new REQ packets and those REQ packets being forwarded. Each TIMBA node originates new REQ packets following the time interval policy, but it is important to note that REQ forwarding at each node does not follow time interval policy. If the node needs to forward an REQ packet, it just does the forwarding (after an autonomous random delay). This is because REQ forwarding is merely a means to let an REQ packet reach every network member. Adversary already knows this feature *a priori*. Since all REQ, whether real or dummy, are flooded and forwarded approximately once on each link, each link capacity is thus of the order $O(N)$. This large link capacity requirement is not needed if dummy REQs are not flooded (as specified in our practical routing scheme ANODR).

Note that this personal trapdoor key is *not* shared with anybody else. The node keeps this truly random key as its personal secret. As illustrated later in REP control flow, here we can use a more efficient trapdoor one-way permutation family f in a symmetric key encryption system[7]. For example, we can use AES and onion length is always 128 bits, a very efficient choice in both computation and communication.

The "one-time public key" field is used in per-hop key agreement. It is from a temporary public/private key pair *generated on the fly* at each REQ forwarding node. One such key pair must be generated per route discovery. It is uniformly distributed over all key pair candidates in a predefined number field (e.g., El Gamal uses $G(q)$ where q is a strong prime). The node stores the temporary private key in the $private_{me}$ field of the connection's soft-state. Then the node can decrypt a reply encrypted by the temporary public key, which is stored in the REQ-downstream node's $public_{upstream}$ field.

Finally when an REQ packet arrives at the real recipient, there are $h_x < (2{\cdot}h_D - 1)$ cryptographic trapdoors embedded in the onion component, where h_x is the number of hops traveled by the REQ packet. The REQ forwarding process is scalable to the number of network members N because this quantity is of a lower order $O(log_2 N)$.

Multi-hop REP control flow: Upon receiving the first REQ packets upon a sequence number (i.e., a unique global trapdoor value), the real recipient wait for $(2{\cdot}h_D - 1)$ time intervals, then it randomly selects at least one of all received REQ packets to reply, in REP format given below:

[7] In theory, Goldreich-Levin hardcore bit is needed in onion construction. This way, the onion construction procedure is identical to Blum-Micali pseudorandom generator (Appendix A.1, Definition 10) where the seed is the random onion-core selected by the source, and the truly random key $personal\_trapdoor\_key$ is used as the truly random nonce used in Goldreich-Levin hardcore bit to generate a new onion in the form of cryptographically strong pseudorandom bits, which are indistinguishable from truly random bits by a $\mathcal{BPP}$ adversary. For the ease of presentation we spare the hardcore bit to illustrate the onion construction procedure in a more intuitive way—simple encryption and decryption.

	← encrypted per-hop using the recorded temporary public key $public_{upstream}$ →
REP (2 bits)	onion (fixed length in a specific public key cryptosystem, e.g., 1024 bits)

For the ease of presentation, let's study the problem assuming the onion field is not encrypted with per-hop key exchanged on the fly. At the real recipient, the onion in the outgoing REP packet is copied from the chosen REQ packet. At the recipient end and also every stop, the REP packet is locally broadcast to all r neighbors. During the local broadcast procedure, the REP packet is encrypted using different encryption keys shared with different neighbors. Now the real forwarder who embedded the outmost trapdoor during REQ phase is in the one-hop broadcast neighborhood. Only this node can inverse the trapdoor one-way function and peel off one layer of the onion. This "broadcast with anonymous trapdoor assignment" procedure is repeated until the original onion-core is returned to the real sender (who should remember and recognize what it has sent). As the trapdoor-based onion structure goes back along the previous forwarding path from the real recipient to the real sender like a boomerang, it is named *trapdoored boomerang onion (TBO)*. In TBO, we can simply use high-speed symmetric key cryptosystems (Figure 6.6). As shown later in Section 6.5, this important feature is critical to boost routing performance on resource-limited mobile nodes.

However, if the onion field is not (re-)encrypted per-hop, then traffic analysts can trace the route by matching the two onions: the one in REQ and the one in corresponding REP. This is the reason why a per-hop key must be used in a semantically secure encryption to randomize the onion field. In practice, the public key can be used to exchange a symmetric key between two nodes of each hop. For example, the REP transmitter chooses a random per-hop symmetric key, then use the one-time public key of its REQ-upstream (i.e., now it is the REP-downstream) node in a semantically secure encryption, so that the REP-downstream node can decrypt the symmetric key.

$$TBO_A = \underline{f(K_A, core)}$$

$$TBO_B = \underline{f(K_B, \underline{f(K_A, core)})}$$

$$TBO_C = \underline{f(K_C, \underline{f(K_B, \underline{f(K_A, core)})})}$$

$$TBO_D = \underline{f(K_D, \underline{f(K_C, \underline{f(K_B, \underline{f(K_A, core)})})})}$$

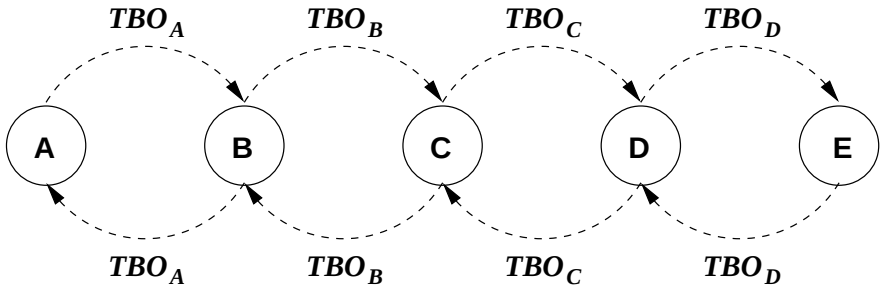

Figure 6.6: **Trapdoored Boomerang Onion (TBO) between source sender A and destination recipient E**

This symmetric key is then used as the per-hop encryption key. REP packet format with such a symmetric key agreement is given below:

	encrypted per-hop using the recorded ← temporary public key $public_{upstream}$ →	per-hop Vernam Cipher using ← the agreed symmetric key →
REP (2 bits)	per-hop symmetric key agreement (fixed length)	onion (fixed length)

During REP phase, the dynamic key exchange scheme is not vulnerable to man-in-the-middle attacks. In other words, an adversary cannot insert itself between two REP forwarding nodes because any onion is formed at the REQ phase and only the destination has the authority to choose which onion is used to establish an on-demand route. The REP forwarding nodes are determined by the onion — those who formed the destination-selected onion during REQ phase are now REP forwarding nodes. The

only chance to launch man-in-the-middle attack is during REQ phase, but such an adversary is indeed an REQ forwarder[8].

As to dummy REP traffic, each TIMBA node sends out dummy REP packets following the time interval policy. It uses a truly random public key.

Multi-hop data flow: Like PipeNet [33], Onion Routing [118], and Freedom [146], data packets in standard TIMBA are forwarded in an efficient manner similar to virtual circuits [7]. This anonymous data forwarding scheme is already described in details in Chapter 5. The per-hop encryption key exchanged on the fly can be used as the secret seed to generate pseudorandom route pseudonym sequences.

Case I: The conceptual data packet format transmitted in a wired standard TIMBA network is given below:

	per-hop Vernam Cipher $\leftarrow$ using the agreed per-hop key $\rightarrow$	
DATA (2 bits)	dummy flag (1 bit)	message payload (fixed length)

To send out dummy traffic, the sender sets the dummy flag to 1.

Case II: The conceptual data packet format transmitted in a wireless standard TIMBA network is given below:

[8] An interesting example is that an adversary forwards the onion field without changing it. This is identical to using the identity function (i.e., $f(x) = x$) as a fake trapdoor one-way function. This adversary is nevertheless an REQ forwarder. Once the onion produced by such REQ forwarders is selected by the destination, the forwarders are en route. They can disrupt ongoing traffic, but cannot compromise mobile anonymity. Such traffic disruption adversary is active adversary (i.e., not the "honest-but-curious" adversary studied in this chapter) that will be addressed in Chapter 7.

	cryptographically strong ← pseudorandom ensemble →	per-hop Vernam Cipher ← using the agreed per hop key →
DATA (2 bits)	route pseudonym (fixed length)	message payload (fixed length)

To send out a dummy data packet, the sender uses a truly random route pseudonym that is out-of-sequence of any route pseudonym sequences in use.

Route error report: To cope with network dynamics, on-demand anonymous connections are refreshed every T_{state}. In addition, if route outage is detected, a node can send an error report packet to its upstream node, which then forwards the error report further toward the upstream. All soft-state status related to the reported connections is recycled on all ERR forwarding nodes. This design implements real time reaction to route outage.

Case I: The conceptual ERR packet format transmitted in a wired standard TIMBA network is given below:

	per-hop Vernam Cipher ← using the agreed per hop key →	
ERR (2 bits)	dummy flag (1 bit)	connection sequence number (fixed length)

Case II: The conceptual ERR packet format transmitted in a wireless standard TIMBA network is given below:

	cryptographically strong ← pseudorandom ensemble →	per-hop Vernam Cipher ← using the agreed per hop key →
ERR (2 bits)	route pseudonym (fixed length)	connection sequence number (fixed length)

To send a dummy ERR packet, a node may either use the same techniques used in dummy DATA packet transmission, or simply embeds a non-existing connection sequence number in the packet.

Security analysis against external adversary: Standard TIMBA ensures perfect mobile anonymity against a $\mathcal{BPP}$ traffic analyst that is also capable of timing analysis. If there are internal adversaries, our claims are applicable to the end-to-end connection events forwarded only by the uncompromised network community.

We use the fact that a $\mathcal{BPP}$ adversary cannot invert one-way functions, and cannot differentiate cryptographically strong pseudorandom ensembles (CSPREs) from truly random ensembles. Thus CSPREs are treated as truly random bits in the analysis. In particular, given two blocks of CSPRE bits of identical size, a $\mathcal{BPP}$ adversary sees two independently chosen random blocks.

We first prove some useful theorems to be used later in the analysis.

Theorem 1 *In standard TIMBA, any two packets of different types cannot be correlated by a $\mathcal{BPP}$ timing analyst.*

Proof: Standard TIMBA has 4 packet types. This gives 6 combinations.

- REQ vs. REP: With or without an incoming REQ a node would send out an REP due to the time interval policy. Only the real recipient's decryption (opening the global trapdoor) can correlate them together. This requires the $\mathcal{BPP}$ analyst to invert the global trapdoor or to compromise the real recipient. Both are infeasible for a $\mathcal{BPP}$ timing analyst.

- REQ/REP/DATA vs. ERR: ERR is reported upon random route outage. The occurrence of an ERR packet is independent from all other types of packets.

Although real ERR packets clear forwarding table entries en route, a $\mathcal{BPP}$ adversary is incapable of knowing this unless it can distinguish CSPRE held in real ERR packets from truly random bits held in dummy ones.

- REQ/REP vs. DATA: With or without control traffic, data traffic is transmitted due to the time interval policy, and vice versa. Only the soft-state information on uncompromised nodes can identify the relation between REQ/REP and DATA.

Therefore, any two packets of different types cannot be correlated together by a $\mathcal{BPP}$ timing analyst. □

Theorem 2 *In standard TIMBA, any two packets of same type but different virtual circuits cannot be correlated by a $\mathcal{BPP}$ timing analyst.*

Proof: A virtual circuit in standard TIMBA is established by REQ/REP route discovery identified by a (computationally) unique global trapdoor.

- Any two REQ packets of different virtual circuits are indeed related only if route outage or timeout T_{state} happens. However, this real relation is only known to the real sender who pumps out dummy REQ following the time interval policy. The $\mathcal{BPP}$ timing analyst cannot identify the correlation unless it can differentiate the two correlated real REQ packets from dummy REQ packets.
- Similarly, the $\mathcal{BPP}$ timing analyst cannot identify two correlated real REP/ERR/DATA packets from dummy REP/ERR/DATA packets. □

Theorem 3 *In standard TIMBA, any two packets of the same type and same virtual circuit cannot be correlated by a $\mathcal{BPP}$ timing analyst.*

Proof: In standard TIMBA, REQ forwarding is merely a preparation phase of virtual circuit establishment. A virtual circuit is not realizable until the moment the destination opens the global trapdoor and selects the reply onion.

We need to prove that two packets forwarded on consecutive hops of a REP/DATA/ERR flow looks truly random (hence independent) from a $\mathcal{BPP}$ adversary's view. Two packets on non-consecutive hops have no direct relation except in the sense they are correlated by packet forwarding on consecutive hops.

We prove the statement by studying content correlation attack and timing analysis.

A $\mathcal{BPP}$ adversary can correlate packet contents by measuring packet size or inspecting data field.

- In terms of packet size, each type of packet (REQ, REP, ERR, DATA) is of uniform size and format, thus the adversary cannot correlate any two packets of the same type by measuring size.
- The 2-bit *type field* identifies four types of packet — REQ, REP, ERR, DATA. But all packets of the same type appear uniformly in the network due to the time interval policy and broadcast policy. The adversary cannot correlate any two packets of the same type by only inspecting the type field.
- The payload area of REP is produced by trapdoor one-way permutation from a one-time truly random key. Permutation does not change the truly randomness. In the alternative REP format with key agreement field, the payload has one more field holding cryptographically strong pseudorandom ensembles, it is indistinguishable from truly random ensembles by the adversary.
- The payload fields of DATA packet and ERR packet are all cryptographically strong pseudorandom ensembles. They look truly random from a $\mathcal{BPP}$ adversary's view.

To trace a real packet flow, a timing analyst must thwart the time interval policy by distinguishing real traffic from dummy traffic. This requires differentiating cryptographically strong pseudorandom ensembles from truly random ensembles. This is beyond the capability of a $\mathcal{BPP}$ adversary. □

Corollary 4 *In standard TIMBA, any two transmission events cannot be correlated by a $\mathcal{BPP}$ timing analyst.* □

Based on the above theorems, we study how standard TIMBA ensures mobile anonymity.

Firstly, perfect *sender identity anonymity*, perfect *recipient identity anonymity*, and perfect *sender-recipient identity relationship anonymity* are ensured because node identity is never used. Let $H(I)$ be the logarithm of the number of uncompromised nodes. $H(I_s|X) = H(I_r|X) = H(I)$ is always true given any intercepted transmission event.

Secondly, we study *sender venue anonymity*. The adversary can break sender venue anonymity by two means: (1) concluding that a real global trapdoor is real, (2) breaking sender-recipient venue relationship anonymity (i.e., tracing a virtual circuit to the sender's venue).

- A $\mathcal{BPP}$ adversary cannot invert a trapdoor one-way permutation to check the contents in a global trapdoor. In other words, it cannot see the $dest$ string and conclude that the global trapdoor is not a dummy.
- Sender-recipient venue relationship anonymity is proven in Theorem 3.

Then we study *recipient venue anonymity*. The adversary can break recipient venue anonymity by two means: (1) concluding that a global trapdoor is encrypted with the

real recipient's key, (2) breaking sender-recipient venue relationship anonymity (i.e., tracing a packet flow to the recipient's venue).

- The global trapdoor is appended with a truly random nonce and encrypted by a trapdoor one-way permutation. The ciphertext is also a random nonce. A $\mathcal{BPP}$ adversary cannot correlate it with anybody's key.
- Sender-recipient venue relationship anonymity is proven in Theorem 3.

Finally we study *location privacy* and *motion pattern privacy*. All models and schemes presented in this dissertation achieve weak location privacy and weak motion pattern privacy because node identity is never revealed. In addition, Corollary 4 has proven that a $\mathcal{BPP}$ timing analyst cannot correlate any pair of transmission events. Thus TIMBA also ensures strong location privacy and strong motion pattern privacy.

Security analysis against internal adversary: Since node identity is never used and revealed, no internal adversary can compromise sender/recipient identity anonymity and sender-recipient identity relationship anonymity of uncompromised nodes.

Both sender and recipient venue anonymity are compromised if sender-recipient venue relationship anonymity is compromised. In addition, to compromise sender venue anonymity of an uncompromised node, the adversary must compromise the real recipient and record the venue where the corresponding global trapdoor value originated; To compromise recipient venue anonymity of an uncompromised node, all neighbors of the node must be intruded so that the adversary can see the node is actually sending back a real reply.

Then we study how passive internal adversary can degrade sender-recipient venue relationship anonymity. A common attack is exploring the revealed soft-state tables.

In order to unlink a network member's identity and its standing location, TIMBA

employs a very different approach from common routing schemes. As depicted in Figure 6.7, common routing schemes use node's identity to furnish packet forwarding, while TIMBA uses an on-demand route discovery process to randomly name each transmission hop and to record the mapping between consecutive hops in each forwarding node. TIMBA's approach bears resemblance to virtual circuits used in Internet QoS [7]. However, the design goal of TIMBA is very different from virtual circuits: When node intrusion occurs in hostile environments, the damage is localized in TIMBA.

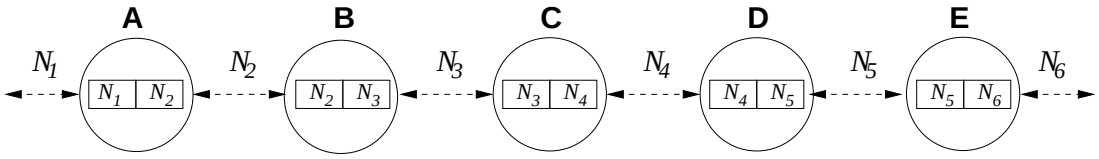

Figure 6.7: **Different approaches in packet forwarding** (Using node pseudonyms $A, B, \ldots$ vs. using route pseudonyms $N_1, N_2, \ldots$)

If a node X is compromised, from the revealed forwarding table the internal adversary can link two random pseudonyms together for each route going through the node X. For each route, if F forwarding nodes are compromised and they are consecutive en route, then a route segment of $F + 1$ hops are linked together. If the compromised nodes are not consecutive en route, then the adversary can form multiple route segments, but it is hard to link together the multiple compromised segments. For example, if A is the source and E is the destination in Figure 6.7, and A, B, D, E are intruded, then adversaries can form traceable segments $\overline{ABC}$ and $\overline{CDE}$, but they have to intrude C to discover that $\overline{ABC}$ and $\overline{CDE}$ belong to the same route.

Let's quantify the damage caused by node intrusion. Suppose the route totally has L hops, K compromised route segments, and the hop count of i-th compromised

segment is F_i,1≤i≤K, we define the *route pseudonym traceable ratio* R of the route as

$$R = \frac{\sum_{i=1}^{K}(F_i \cdot W_i)}{L} = \frac{\sum_{i=1}^{K}(F_i \cdot \frac{F_i}{L})}{L}$$

where W_i is a weight factor. Without loss of generality[9], we select $W_i = \frac{F_i}{L}$ so that the traceable ratio of a route is 100% when all forwarding nodes en route are intruded, or 0 when no forwarding node en route is intruded. In addition, the longer a compromised segment is, the larger the traceable ratio R is as the adversary can trace a longer distance towards its target. Using the same example from the previous paragraph, $L = 4$. The traceable ratio $R = \frac{2 \cdot \frac{2}{4} + 2 \cdot \frac{2}{4}}{4} = \frac{1}{2}$ when A, B, D, E are intruded, or $R = \frac{3 \cdot \frac{3}{4} + 1 \cdot \frac{1}{4}}{4} = \frac{5}{8}$ when A, B, C, E are intruded.

In terms of soft-state table revelation, perfect sender-recipient venue relationship anonymity is ensured if the traceable ratio between a sender and a recipient is 0. Sender-recipient venue relationship anonymity is completely compromised if the traceable ratio is 1.

But there are more subtle internal attacks to be studied. Let's use tagging attack [34] as an example. In tagging attack, an active internal adversary en route modifies a data payload by embedding a pattern (for example by flipping a bit) so that the pattern can be recognized later by another active internal adversary en route (who may restore the packet content to disguise the intrusion). Although they don't know how the packet is routed between them, they know they must be on the same packet flow. Such active attacks will be studied in Chapter 7.

Passive internal attacks can apply a similar strategy. Two or more passive internal adversaries can collaborate and compare message payload contents. We may let the sender (source) share an encryption key with each forwarder, and adding Chaum's

[9] The weight W_i can be of form $(\frac{F_i}{L})^r$ where $r \geq 0$.

sender-centric encryption design (i.e., each message payload is an onion). But this does not help too much, since the sender/source could be such an internal adversary who wants to trace a destination victim by initiating the traceable packet flow. This internal attack differentiates end-to-end message payload from routing information — while per-hop route information (e.g., route pseudonyms) can be established on-demand and independently chosen, application message payload is by its nature a static set of information between the two communicating ends — the recipient must receive the same message payload the sender knows. Therefore, no matter which nodes are responsible for (re-)encryption or (re-)decryption for each forwarder, application message payload can be matched by internal attackers if the responsible encrypters/decrypters are compromised.

Cost complexity: In a hypercube network of N nodes, two cases are possible: (1) If TIMBA nodes send out dummy REQ packets according to the time interval policy, then each TIMBA node sends/receives $(3 + O(N) + d)$ packets to each neighbor per time interval, where the $O(N)$ complexity is caused by REQ packet forwarding. (2) If TIMBA nodes do not send out dummy REQ packets, then each TIMBA node only sends/receives $(2+d)$ or $(3+d)$ packets to each neighbor per time interval. The second case trades sender venue anonymity for performance.

On the other hand, the network diameter $h_D = r = log_2 N$ is the maximal distance between any two nodes. An anonymous connection can be established within the bounded round-trip latency

$$T_{int} \cdot 2 \cdot h_D = T_{int} \cdot 2 \cdot log_2 N.$$

After an anonymous connection is established, a data message of l packets long

will be delivered from a sender to its recipient within the bounded latency

$$T_{int} \cdot \left\lceil \frac{l}{d} \right\rceil \cdot h_D = T_{int} \cdot \left\lceil \frac{l}{d} \right\rceil \cdot log_2 N.$$

6.2 Let wireless broadcast help anonymity

If the anonymity models are implemented in a conceptual network with wired point-to-point links, then broadcast (either global in TIBA or local in TIMBA) is realized through replication over multiple point-to-point transmissions, i.e., sending r different packets to r different neighbors. This is not true in mobile wireless networks using omnidirectional radio links.

We seek to explore two readily available broadcast mechanisms in mobile wireless networks: (1) The first one is one-hop omnidirectional wireless broadcast radio. Once a wireless sender transmits a signal, the signal is received by all one-hop neighbors in a TIMBA network. Real recipient is hidden inside the one-hop recipient group. (2) The second one is multi-hop network flooding of a route request message. In TIMBA, this feature ensures that every network member can be the real recipient, hence also effectively hides the real recipient in the entire network.

With the help from these two wireless broadcast mechanisms, we will illustrate in the next section through the design of ANODR, a practical multi-hop anonymous routing scheme for mobile wireless networks, on how to achieve mobile anonymity following the TIMBA model and how to trade off security guarantee with routing performance.

6.3 ANODR: practical anonymous routing for multi-hop wireless networks

The standard TIMBA has all critical features we need, for example, on-demand approach, one-hop anonymous data forwarding using route pseudonyms, etc., but it is an ideal model that may not be realized efficiently. ANODR is a compromise between the ideal TIMBA model and performance concerns raised in the real world. ANODR has three variants differing in their performance.

1. The first one is an "anonymous-only" routing protocol that ensures *sender identity anonymity*, *recipient identity anonymity*, *sender-recipient identity relationship anonymity*, and *weak location privacy* for mobile nodes. Its performance is comparable to common on-demand routing protocols in use since it relies on high-speed symmetric key cryptosystems.

2. The second variant, "anonymous+untraceable ANODR", in addition, provides anonymity supports on *recipient venue anonymity*, *sender-recipient venue relationship anonymity*, *strong location privacy*, and *strong motion pattern privacy*. This variant is the nearest approximation of standard TIMBA. The major difference between this variant and standard TIMBA is that the time interval policy is not enforced on REQ traffic, thus sender venue anonymity can be compromised by timing analysis. Besides, the time interval policy is replaced by "neighborhood traffic mixing" on REP/ERR/DATA traffic. Traffic mixing is a practical but not perfect implementation of the time interval policy.

3. The third variant, "anonymous+untraceable ANODR-KPS", uses Key Pre-distribution Schemes (KPS, for example, Blom's scheme, see Appendix A.2) to establish per hop encryption keys used in anonymous connections. It provides the

same set of anonymity supports as in “anonymous+untraceable ANODR”, but only *weak location privacy* and *weak motion pattern privacy* are supported due to applying static pseudonyms in key agreement.

The reason for the need is of the tradeoffs in routing performance and security guarantee. The two “anonymous+untraceable ANODR” variants pay more cost to stop content & causality correlation attacks. In particular, “anonymous+untraceable ANODR-KPS” features better performance because it does not use expensive public key cryptography in anonymous connection establishment. The tradeoffs between public key cryptography and KPS are:

- In a key agreement scheme based on public key cryptography, the secrecy of exchanged keys is not affected by other network members’ activities. And the key generation module can generate one-time public/private key pair upon request. But such a scheme is expensive in computation.
- In a key agreement scheme based on KPS, computation is efficient. However, the network can only tolerate certain number of node compromises. The entire scheme is compromised once the adversary has compromised more than the threshold number of nodes. In addition, a KPS scheme cannot randomize its pre-loaded key agreement materials.

6.3.1 Practical network assumptions

We assume the network is comprised of all kinds of heterogeneous nodes with very different computational resources as well as diverse roles in a covert operation. Nevertheless, all nodes use the same addressing system, e.g., 32-bit IPv4, 128-bit IPv6, or equivalence.

At physical layer, wireless links are symmetric and omnidirectional[10]; that is, if a node X is in transmission range of some node Y, then Y is in transmission range of X. At link layer, a node's medium access control (MAC) interface is capable of physically broadcasting data packets locally. Within its transmission range, a network node can use physical broadcast to send a unicast packet to a specific node, or a broadcast packet to all local nodes. By anonymous acknowledgment and re-transmission, a local sender and a local receiver can implement locally reliable unicast. If the count of re-transmission exceeds a predefined threshold, the sender considers the connection on the hop is lost. Finally, in this dissertation we focus on data forwarding (link layer) and routing (network layer). We do *not* cover untraceability problems at the physical layer or the application layer. It is beyond the scope of this dissertation to study how to trace a network node using signal delay and triangulation at the physical layer.

Scalability is a critical issue in practical networking. It is well known that wireless communication using omnidirectional broadcast radio is not scalable to the number of hop counts in an average connection [58][88]. However, Li et al. [88] also showed that scalability can be achieved if communication pattern is localized. Here we interpret this conclusion in terms of the hypercube structure. That is, if communication pattern can be localized into a hypercube structure of N nodes where average node density is $O(\log N)$, then the hop count of an average connection is also $O(\log N)$. Then the communication scheme is potentially scalable due to the mathematic properties of hypercube. When number of network members increases linearly in $O(N)$, we suggest to use longer range radio so that average hop count only increases logarithmically in $O(\log N)$. This design direction opens more challenges in link layer protocol design

[10] Here we discuss an 802.11-like wireless MAC scheme. Though non-broadcasting wireless MAC schemes (such as directional antenna technology) are under development, broadcast based MAC continues to be an affordable solution that can be used by all network nodes.

that is beyond the scope of this dissertation. Exploring the relation between scalability and mobile anonymity is our critical future work to pursue.

For practical key agreement, we assume network members are pre-loaded with *a priori* key materials. If public key cryptography is used, the key assumption is stated in TIBA model. If KPS is used, we assume that an offline key pre-distribution dealer can send personal key information to every network member via a private channel. Each network member must obtain such private personal key information before joining the network.

6.3.2 Design rationales

Tradeoff between ideal model and real world TIMBA consumes massive bandwidth resources and incurs unacceptable communication latency if uniform traffic pattern is implemented over every time interval. In particular, since one REQ will cause a network-wide flooding in on-demand routing schemes, the time interval policy is *not* enforced on ANODR's control traffic (i.e., no dummy REQ). In addition, ANODR uses on-demand *neighborhood traffic mixing* to approximate rather than realize uniform data traffic.

Protect location privacy and motion pattern privacy for mobile networks No adversary, including internal adversary, can correlate anything with any non-compromised network member's identity. Weak location privacy and motion pattern privacy are ensured in all ANODR variants. This is possible due to two design features: (1) Rather than using traditional node-based routing schemes, TIMBA/ANODR's routing is realized by naming each hop on a multi-hop data forwarding path with a route pseudonym. (2) Based on modern key agreement schemes, pre-distributed *a priori* key materials can establish pairwise key between any two network members. TIMBA/ANODR uses such pairwise key agreement to set global and local trapdoors in establishing ano-

nymous connections. As a result, to our knowledge ANODR is the first routing protocol where node identities are not needed at all (thus never used and compromised). This is different from other routing protocols using route pseudonyms: ATM uses VCI, but it needs node identities in establishing virtual circuits; Onion Routing uses ACI, but it needs the names of all Onion Proxies to establish interconnections among those Proxies. In a nutshell, so far no routing protocol other than ANODR provides location privacy and motion pattern privacy supports.

Dissociate mobile anonymity from legacy content privacy In our design, anonymous routing in mobile wireless networks is orthogonal to legacy content privacy. Network members may employ end-to-end security protocols (e.g., SSL/TLS, host-to-host IPsec) to ensure privacy of their application payloads. Such protocols provide security services at or above the network layer, and are not the subjects studied in this work.

Intrusion tolerant design In hostile environments, intrusion is likely inevitable over a long time window. A distributed protocol vulnerable to single point of compromise is not a proper solution. A qualified solution should maximize its tolerance to multiple compromises, especially against passive internal adversaries who exhibit no malicious behavior and stay in the system. The number of such passive internal adversaries can add up to non-trivial amount over a long time interval.

6.3.3 Design details of "anonymous-only ANODR"

ANODR divides the routing process into two parts: *anonymous route discovery* and *anonymous route maintenance*. Besides, in *anonymous data forwarding* data pack-

ets marked with random route pseudonyms are routed anonymously from senders to receivers. The details of these parts are described below:

Anonymous route discovery Anonymous route discovery is a critical procedure that establishes random route pseudonyms for an on-demand route. A communication source initiates the route discovery procedure by assembling an RtREQ packet and locally broadcasting it. The RtREQ packet is of the format

$$\langle RtREQ, seq\# = tr_{dest}, TBO \rangle,$$

where (i) $seq\#$ is a computationally unique sequence number generated by one-way hash function from the input $\langle dest, nonce \rangle$, where $dest$ is the special string and $nonce$ is a random nonce. (ii) $onion$ is a 128-bit TBO using only high-speed symmetric key cryptosystems.

How ANODR protocol establishes a multi-hop anonymous path is described below.

1. *RtREQ phase*: RtREQ packets with previously seen sequence numbers are suppressed. Otherwise, as depicted in Figure 6.6, each RtREQ forwarding node X uses a random symmetric key K_X to embed a trapdoor to the TBO. This is done by $TBO_{out} \longleftarrow f(K_X, TBO_{in})$ where f is a trapdoor one-way permutation with a symmetric trapdoor key. Then the modified RtREQ packet is broadcast locally. The secret information $(TBO_{in}, K_X, TBO_{out})$ is only known to X.

2. *RtREP phase*: When the destination receives an RtREQ packet, the embedded TBO can be used to establish an anonymous route towards the source. The destination opens the global trapdoor and assembles an RtREP packet of the format

$$\langle RtREP, N, pr_{dest}, TBO \rangle$$

where pr_{dest} is the anonymous proof of global trapdoor opening, and N is a locally unique random route pseudonym. The RtREP packet is then transmitted by local broadcast.

After each local RtREP broadcast, only the next hop (i.e., the previous hop in RtREQ phase) can covertly open the trapdoor it made in the RtREQ phase, hence the result is equivalent to an anonymous wireless unicast. Then the forwarder strips a layer of the TBO, selects a locally unique nonce N', stores the correspondence between $N \rightleftharpoons N'$ in its forwarding table, peels off one layer of the onion, replaces N with N', then locally broadcasts the modified RtREP packet. The same actions will be repeated until the source receives the "onion core" it originally sent out.

Upon receiving different RtREQ packets, the destination can initiate the same RtREP procedure to establish multiple anonymous paths between itself and the source. ANODR leaves the decision to be made by implementation defined policies.

Anonymous route maintenance Following a soft state design, the routing table entries are recycled upon a predefined timeout T_{win}. Moreover, when one or more hop is broken due to mobility or node failures, nodes cannot forward packet via the broken hops. We assume nodes can detect such anomalies when re-transmission count exceeds a predefined threshold. Upon anomaly detection, a node looks up the corresponding entry in its forwarding table, finds the other route pseudonym N' which is associated with the pseudonym N of the broken hop, and assembles a route error packet of the format $\langle RtERR, N' \rangle$. The node then recycles the table entry and locally broadcasts the RtERR packet.

A receiving node of the RtERR packet looks up N' in its forwarding table. If the

lookup returns result, then the node is on the broken route. It should find the matched N'' and follow the same procedure to notify its neighbors.

Anonymous data forwarding For each end-to-end connection, the source wraps its data packets using the outgoing route pseudonym in its forwarding table. A data packet is then broadcast locally without identifying the sender and the local receiver. The sender does not bother to react to the packet it just sent out. All other local receiving nodes must look up the route pseudonym in their forwarding tables. The node discards the packet if no match is returned from its forwarding table. Otherwise, it changes the route pseudonym to the matched outgoing pseudonym, then broadcasts the changed data packet locally. The procedure is then repeated until the data packet arrives at the destination.

Setting and opening global trapdoor Let Fig. 6.6 be the example scenario. Initially source A only knows its destination E's certified public key PK_E. Only for the first-time route request, the complete RtREQ format is:

$$\langle RtREQ, tr_{dest} = \{dest, K_{reveal}, K_{AE}\}_{PK_E}, \ K_{reveal}(dest), TBO\rangle$$

where the random K_{AE} selected by A can only be decrypted by E and will be used in later route requests as a shared symmetric key. That is, the complete RtREQ format for all later route requests is:

$$\langle RtREQ, tr_{dest} = K_{AE}(dest, K_{reveal}), K_{reveal}(dest), TBO\rangle$$

RtREP format is: $\langle RtREP, N, pr_{dest} = K'_{reveal}, TBO\rangle$.

Among all network members, only destination E can see the special tag $dest$ and conclude it is the intended destination. The value K_{reveal} is a trapdoor commitment

value. It is a committed secret during RtREQ phase, but will be revealed during RtREP phase. The destination E must present this commitment value $K'_{reveal} = K_{reveal}$ to prove that it has successfully opened the global trapdoor. Any forwarding node can verify the anonymous proof of global trapdoor opening by checking $K_{reveal}(dest) \stackrel{?}{=} K'_{reveal}(dest)$. Other nodes other than the destination E cannot show the correct K_{reveal} unless it can break encryption.

6.3.4 Design details of "anonymous+untraceable ANODR" and "anonymous+untraceable ANODR-KPS"

The ANODR protocol described above only employs high-speed symmetric key cryptosystems, thus is very efficient. However, even though no node identity is revealed during routing, adversaries can trace ad hoc routes to sources and destinations, for example, via content correlation and causality correlation.

Content correlation against RtREQ/RtREP and data flows Matching onions or pr_{dest} in different RtREQ/RtREP packets is an example of content correlation: (1) Even assuming an ideal uniform traffic pattern, recorded TBOs from RtREQ packets can be used to identify an ad hoc route—TBOs appear again in RtREP packets are used by ANODR to establish ad hoc routes. (2) RtREP packets with the same pr_{dest} field belong to the same path.

Case I: "Anonymous+untraceable ANODR" employs one-time public key exchange at real time to thwart content correlation attacks. During the RtREQ phase, a forwarding node must append its one-time public key from a temporary public/private key pair

(pk_{one}, sk_{one}). RtREQ packet format is[11]:

$$\langle RtREQ, tr_{dest}, TBO, pk_{one} \rangle,$$

and RtREP packet format is:

$$\langle RtREP, \{K_{seed}\}_{pk_{one}}, K_{seed}(pr_{dest}, TBO) \rangle,$$

where K_{seed} is a nonce key (same as the pseudonym N in the anonymous-only AN-ODR). K_{seed} will function as a shared secret between the two ends of the hop. During the RtREP phase, each stop can covertly use its own one-time sk_{one} to recover the needed information as usual. This design ensures that there is no expensive public key computation incurred during RtREQ flooding. During the RtREP phase, each forwarding node en route must do one encryption and one decryption using the well-known public key scheme.

For the one-time public/private key pair, storage overhead can be traded off for key generation overhead as the node may generate a number of such key pairs prior to joining in the ad hoc network. In addition, the key length should be minimized to reduce transmission overhead, but must be long enough to resist cryptanalysis. AN-ODR recommends elliptic curve based schemes, such as ECAES, with key length ranging from 112-bit to 160-bit (approximately equivalent to RSA using 512-bit to 1280-bit key length [87]) to resist a 1-day cryptanalysis with hardware cost ranging from $50,000,000 to $250,000,000.

Case II: "Anonymous+untraceable ANODR-KPS" employs KPS key exchange at real time to thwart content correlation attacks. During the RtREQ phase, a forwarding node

[11] For the ease of presentation, we continue to use tr_{dest} and pr_{dest} in our notation. The instantiation of these notions is described in "setting and opening global trapdoor".

must append its key agreement information. RtREQ packet format is:

$$\langle RtREQ, tr_{dest}, TBO, \underset{\sim}{key\_agreement\_info\_for\_K_{seed}} \rangle,$$

and RtREP packet format is:

$$\langle RtREP, \underset{\sim}{K_{seed}}(pr_{dest}, TBO), \underset{\sim}{key\_agreement\_info\_for\_K_{seed}} \rangle,$$

where either side can determine the agreed K_{seed} based on exchanged key agreement information. Using Blom's scheme as an example, the key agreement information is the corresponding column in the public matrix (Appendix A.2). Nevertheless, a KPS scheme cannot randomize its key agreement information like the usage of *one-time* public keys. This means strong location privacy and motion pattern privacy are compromised because such static key agreement information is actually a static pseudonym of the node—REQ and REP packets from the same node can be correlated together. We understand this situation and KPS is used in this work merely for boosting routing performance, not for security claims.

Besides RtREQ/RtREP flows, data packet flows are also vulnerable to content correlation. (1) In 802.11, the shared secret K_{seed} can be used as WEP key to implement data payload (re-)encryption per hop. (2) As in TIMBA, "anonymous+untraceable ANODR" must implement one-time route pseudonyms by using the K_{seed} as the seed to generate a sequence of pseudorandom route pseudonyms to be stamped on data packets; (3) Due to performance concerns, ANODR does not enforce the policy of sending all data packets in a network-wide fixed uniform length—each node just au-

tonomously pads some random bits of random length, the next stop strips off the random padding and adds its own random padding.

Causality correlation against data and RtREP/RtERR flows To thwart timing analysis, ANODR uses *neighborhood traffic mixing*, a method similar to those proposed in various MIX-Net designs [112][76][14]. Let's assume node X autonomously chooses t_X as its playout time window size and r_X as its playout buffer size. During t_X period, if node X has received r data packets with distinct pseudonyms, then it generates $r_d = r_X - r$ random dummy packets. ANODR's mixing is on-demand/reactive as it does not generate dummy packets ($r_d = 0$) if $r = 0$ or $r_X \leq r$. This design is different from TIMBA where each node is required to transmit a certain number of packets for a time interval. The random pseudonyms used in the dummy packets should be out of the synchronization with any pseudonym sequence in use. At the end of time window t_X, the node X randomly re-orders the r_X packets and sends them out in batch.

Unlike a wired link, wireless medium is shared by all local nodes. Thus r is the number of all packets received from the entire one-hop neighborhood during t_X, including those packets not intended for the node. Moreover, since it is useless to inject more dummy packets when the local wireless channel is congested, the threshold ratio $\frac{r_X}{t_X}$ should be set as a value lower than channel bandwidth (e.g., 11Mbps for 802.11b). This neighborhood traffic mixing decreases the chances of sending excessive dummy packets. Nevertheless, any dummy packet would consume significant communication and energy resources, thus ANODR allows each node to trade untraceability with performance. The node X may autonomously shrink the size of its playout time window, or generates less dummy packets to decrease the overhead. Luckily this per-node autonomous traffic mixing policy may help untraceable routing, as a heavy traffic in a locality could be either the result of real traffic or a busy local node with lots of energy to burn and transmitting many dummy packets.

In addition to data packets, RtREP and RtERR packets are also threatened by timing analysis. Similarly, in ANODR each node can *optionally* send dummy RtREP and RtERR packets to confuse the traffic analysts. A dummy RtREP packet uses a random dummy pk_{one} in encryption so that nobody can decrypt it. A dummy RtERR packet uses a random pseudonym that is out-of-synchronization of any pseudonym sequence in use.

6.3.5 Discussions for all ANODR variants

Mobile anonymity support Table 6.2 offers a coarse comparison of mobile anonymity supports provided in MIX-Net, all three ANODR variants, and the ideal TIMBA model.

In the table, the expression "traffic mixing is sound" means a conditional statement. In practice, we know traffic mixing may not completely thwart traffic analysis if it does not implement uniformly distributed traffic pattern [143][36]. Unlike the time interval policy, traffic mixing is merely a practical approach that is regarded as an effective countermeasure against timing analysis.

In addition, "anonymous+untraceable ANODR" only uses limited public key cryptography in RtREP flow. The minimized computational overhead incurred by public key cryptography is smaller than the counterpart overhead incurred by MIX-Net. The critical difference between ANODR variants and the ideal TIMBA model is listed below:

- A typical mobile ad hoc network does not possess the $O(\log N)$ topological complexities of a hypercube structure. It is an open challenge to build a scalable homogeneous ad hoc network [58][59][88]. The on-demand approach used in TIMBA/ANODR faces the challenge of building scalable network protocols.

Table 6.2: **Comparison of mobile anonymity supports**

MIX-Net (if applied in mobile wireless networks & every mobile node is a MIX)	1. No differentiation between *identity anonymity* and *venue anonymity*. 2. *Sender anonymity* ensured if traffic mixing is sound. 3. *Recipient anonymity* not protected against the last forwarder. 4. Supports *sender-recipient venue relationship anonymity* if traffic mixing is sound. 5. *Location privacy* and *motion pattern privacy* not considered. Downstream neighbor's identity always revealed to the current forwarder. 6. Very expensive computational cost due to excessive public key cryptography. 7. Very expensive communication cost due to traffic mixing. 8. Impractical mobile and cryptographic assumptions. 9. *A priori* network topology knowledge stored on each sender compromises mobile anonymity in the presence of a single adversarial sender.
anonymous-only ANODR (public K_{seed}; No per-hop key agreement; No traffic mixing)	1. All *identity anonymity* perfectly ensured. 2. All *venue anonymity* not protected against content & timing analysis. 3. *Weak location privacy* and *weak motion pattern privacy* ensured. 4. Efficient design using symmetric key cryptography and no traffic mixing.
anonymous+untraceable ANODR-KPS (KPS based K_{seed} agreement; Neighborhood traffic mixing)	1. All *identity anonymity* perfectly ensured. 2. Supports *sender venue anonymity* if no adversary in sender's cell. 3. Supports *recipient venue anonymity* if traffic mixing is sound. 4. Supports *sender-recipient venue relationship anonymity* if traffic mixing is sound and not all forwarders en route are adversarial. 5. *Weak location privacy* and *weak motion pattern privacy* ensured. 6. Efficient computational design using symmetric key cryptography. 7. Expensive communication cost due to neighborhood traffic mixing.
anonymous+untraceable ANODR (One-time public key based K_{seed} agreement; Neighborhood traffic mixing)	1. All *identity anonymity* perfectly ensured. 2. Supports *sender venue anonymity* if no adversary in sender's cell. 3. Supports *recipient venue anonymity* if traffic mixing is sound. 4. Supports *sender-recipient venue relationship anonymity* if traffic mixing is sound and not all forwarders en route are adversarial. 5. Supports *strong location privacy* and *strong motion pattern privacy*. 6. Expensive computational cost due to limited public key cryptography. 7. Expensive communication cost due to neighborhood traffic mixing.
standard TIMBA	1. All *identity anonymity* perfectly ensured. 2. Perfect *sender venue anonymity* ensured. 3. Perfect *recipient venue anonymity* ensured. 4. Perfect *sender-recipient venue relationship anonymity* ensured. 5. *Strong location privacy* and *strong motion pattern privacy* ensured. 6. Expensive computational cost due to limited public key cryptography. 7. Impractical communication design due to the time interval policy.

- In all three ANODR variants, the time interval policy is not enforced on RtREQ traffic. This avoids periodic network flooding in ad hoc networks, but sacrifices sender venue anonymity.

- In "anonymous+untraceable ANODR" and "anonymous+untraceable ANODR-KPS", the time interval policy is replaced by neighborhood traffic mixing on RtREP/RtERR/DATA traffic. In neighborhood traffic mixing, no dummy data is sent when there is no real data. This spares every node's transmission energy, but reveals the existence of real transmissions.

- The efficient encryption schemes and pseudorandom generators used in all ANODR variants may not be provably secure. In particular, Blum-Micali pseudorandom generator is too slow to be used in routing.

- In all ANODR variants, DATA packets are not in fixed uniform size. Every node just strips off the random padding from the previous stop, and adds its own random padding. This trades off security with performance.

Reliability of local broadcasts In RtREP/RtERR packet transmission and also in reliable data communication, local broadcasts must be reliably delivered to the intended receiver despite wireless interference. This can be achieved by anonymous acknowledgments. Once the receiver has opened the trapdoor and anonymously received the data, it should locally broadcast an anonymous ACK packet. In an anonymous ACK packet, the source or destination MAC address is the predefined all-1's broadcast address. The packet payload uniquely determine which packet is being acknowledged. In particular, route pseudonyms can be embedded in the ACK's payload to acknowledge an RtREP/RtERR packet or application data packet.

At the other end of the hop, the sender must try to re-transmit data packets until it receives the anonymous acknowledgment. Like 802.11's reliable unicasts, if retrans-

mission count exceeds a predefined threshold, then the node considers the hop connection is broken. If this happens during application data forwarding, route maintenance will be initiated to refresh forwarding table entries.

Routing optimizations One limitation of ANODR is the sensitivity to terminal node mobility. As nodes move, the path is broken and must be reestablished. The well-known AODV and DSR "repair" strategies (which typically benefits from routes cached during unrelated path establishments) cannot be applied here since only anonymous paths specifically set up for the current connection can be used, or the optimization technique by the design conflicts with the anonymity goals.

To enhance performance in a mobile environment, and in particular to mitigate the disruption caused by path breakage, we encourage actual implementations to use multiple paths discussed in the anonymous route discovery part. Several multi-path routing techniques have been described and evaluated in the ad hoc routing literature[106] [86] [91] [104]. Several paths can thus be computed and are used in a round robin schedule. If the application runs on TCP, a TCP protocol resilient to out-of-sequence must be used. Sequential path computation has the advantage of allowing online maintenance—if a path fails, a new path is computed while the remaining paths are still in use.

6.4 Evaluation of cryptographic implementation

In our cryptographic implementation, the length of src, $dest$ tags and route pseudonym (i.e., K_{seed}) nonces is 128-bit. In RtREQ packet, the sequence number $seq\#$ is formed by concatenating the $dest$ string, the current 32-bit clock timestamp on the source, and an arbitrary (but reasonably) long random nonce, then applying encryption to the concatenation.

The processing overhead used in our simulation is based on actual measurement on a low-end device. Table 6.3 shows the performance of different cryptosystems. For public key cryptosystems, the table shows processing latency per operation. For symmetric key cryptosystems (the five AES final candidates), the table shows encryption/decryption bit-rate.

Table 6.3: **Processing overhead of various cryptosystems** (on iPAQ3670 pocket PC with Intel StrongARM 206MHz CPU)

Cryptosystem	decryption	encryption
ECAES (160-bit key)	42ms	160ms
RSA (1024-bit key)	900ms	30ms
El Gamal (1024-bit key)	80ms	100ms
AES/Rijndael (128-bit key & block)	29.2Mbps	29.1Mbps
RC6 (128-bit key & block)	53.8Mbps	49.2Mbps
Mars (128-bit key & block)	36.8Mbps	36.8Mbps
Serpent (128-bit key & block)	15.2Mbps	17.2Mbps
TwoFish (128-bit key & block)	30.9Mbps	30.8Mbps

6.5 Simulation study of ANODR

We implement ANODR in simulation as a basic on-demand route discovery/maintenance scheme with flavors of both source routing and table driven. The source routing part is adopted to simulate the appending and pealing off layers in RtREQs and RtREPs, a way that is similar to the creation and transmission of RtREQs and RtREPs in DSR. The table driven part is used to establish the per hop pseudonym switching during RtREP propagation and data forwarding, a way that is similar to the routing table maintenance in AODV. Possible optimizations used for AODV and DSR are not used in our implementation, for example, no expanding ring search, no local route repair, no promiscuous listening, no salvaging, no gratuitous route repair, no aggressive caching

and no switching entry reuse at intermediate nodes. In addition, ANODR also implements larger RtREQ, RtREP, and RtERR packets with extra processing overhead for encryption and decryption at each packet stop.

We evaluate our proposed routing schemes in three aspects. First, we investigate untraceability of ANODR in terms of intrusion tolerance. As ANODR uses a way similar to source routing in establishing a route, we compare ANODR to DSR. For ANODR, a node intrusion unconditionally exposes everything cached on the node including the mapping between two sets of random route pseudonyms. For DSR, we assume it is protected by an ideal hop-based link encryption scheme. Nevertheless, the entire DSR route will be exposed as long as a packet passing through a compromised node. We use *traceable ratio* R to quantify the effect of node intrusions. The traceable ratio for a DSR route is 0 when none of the nodes en route is intruded, or is 1 otherwise.

Then we evaluate the performance of all ANODR variants: "anonymous-only ANODR", "anonymous+untraceable ANODR" and "anonymous+untraceable ANODR-KPS". In "anonymous+untraceable ANODR-KPS" we have tried two different KPS schemes: one is Blom's deterministic KPS scheme [16] and the other is the probabilistic KPS scheme recently proposed by Du et al. [44]. Du's KPS scheme can be considered as applying Eschenauer-Gligor's probabilistic KPS scheme [46] on top of Blom's deterministic KPS scheme. It features more flexible tradeoffs between security and key agreement probability. In our performance evaluation these two ANODR versions will be denoted as "anonymous+untraceable ANODR-BLOM-KPS" (where Blom's deterministic KPS is used) and "anonymous+untraceable ANODR-DU-KPS"(where Du et al's probabilistic KPS is used). In "anonymous+untraceable ANODR-DU-KPS", the probability of a successful key agreement per hop is 90%, which means during RtREP phase the probability of establishing the anonymous virtual circuit per hop is 90%.

Computational delay using symmetric key cryptosystem AES/Rijndael (approximately 0.02ms for each onion construction) is added to each RtREQ and RtREP forwarding stop. For "anonymous+untraceable ANODR", additional key processing time for RtREP packets ($42 + 160 = 202$ms) is added according to our measurement. The two KPS based schemes trade link overhead for processing time, e.g,"anonymous+untraceable ANODR-BLOM-KPS" uses 1280 bits G-matrix-column exchange information in RtREQ and RtREP packets and "anonymous+untraceable ANODR-DU-KPS" uses 1344 bits and 1288 bits for RtREQ and RtREP respectively. Each of them requires only 1ms extra time in processing RtREP packets. For comparison, AODV with route optimization are also presented in simulation.

Finally, we evaluate the impact of mixing technique on ANODR performance. We study both mixing overhead and routing performance given many combinations of mixing playout window sizes and playout buffer sizes. In the experiment, the dummy packet size is a random value computed from the average size of data packets recently received.

Metrics we used for routing performance include: (i)Packet delivery fraction – the ratio between the number of data packets received and those originated by the sources. (ii) Average end-to-end data packet latency – the time from when the source generates the data packet to when the destination receives it. This includes: route acquisition latency, processing delays at various layers of each node, queueing at the interface queue, retransmission delays at the MAC, propagation and transfer times. (iii) Normalized control packet overhead – the number of routing control **packets** transmitted by a node normalized by number of delivered data packets, averaging over all the nodes. Each hop-wise transmission of a routing packet is counted as one transmission. (iv) Normalized control byte overhead – the total **bytes** of routing control packets transmitted by a node normalized by delivered data bytes, averaging over all

the nodes. Each hop-wise transmission of a routing packet is counted as one transmission. This metric is useful in evaluating the extra control overhead of ANODR. (v) Dummy packet ratio – the ratio between the number of dummy data packets and real data packets given a specific playout time window and buffer size.

6.5.1 Simulation Model

The routing protocols are implemented within QualNet^{TM} [127], a packet level simulator for wireless and wired networks, developed by Scalable Network Technologies Inc. The distributed coordination function (DCF) of IEEE 802.11 is used as the MAC layer in our experiments. It uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets to provide virtual carrier sensing for unicast data packets to overcome the well-known hidden terminal problem. Each data transmission is followed by an ACK. Broadcast data packets are sent using CSMA/CA only. The radio uses the *two-ray ground reflection* propagation model and has characteristics similar to a commercial radio interface (e.g., Lucent's WaveLAN). The channel capacity is 2 Mbits/sec.

In order to hide the sender's and receiver's identity, ANODR's local broadcast with trapdoor uses broadcast address rather than source and destination's link layer addresses. This behavior makes ANODR's transmission look like 802.11 broadcast. However, ANODR's local broadcast with trapdoor is an equivalence of 802.11's unicast rather than broadcast, except that 802.11 uses traceable identity pseudonyms while ANODR uses untraceable trapdoors (with simple table lookup). In data forwarding we use 802.11 unicast plus 1μs table lookup delay to simulate ANODR's local broadcast with trapdoor. We believe it is practical to implement the same feature in commercial 802.11 device drivers.

The network field is 2400m$\times$600m with 150 nodes initially uniformly distributed. The transmission range is 250m. *Random Waypoint* mobility model [73] is used to

simulate nodes' motion behavior. According to the model, a node travels to a random chosen location in a certain speed and stays for a whole before going to another random location. In our simulation, mobility speed varies from 0 to 10 m/sec, and the pause time is fixed to 30 seconds. CBR sessions are used to generate network data traffic. For each session, data packets of 512 bytes are generated in a rate of 4 packets per second. The source-destination pairs are chosen randomly from all the nodes. During 15 minutes simulation time, a constant, continuously renewed load of 5 short-lived pairs is maintained. The simulations are conducted in identical network scenarios (mobility, communication traffic) and routing configurations across all the schemes. Results are averaged over multiple runs with different seeds for the random number generator.

6.5.2 Simulation Results

6.5.2.1 Traceability Analysis

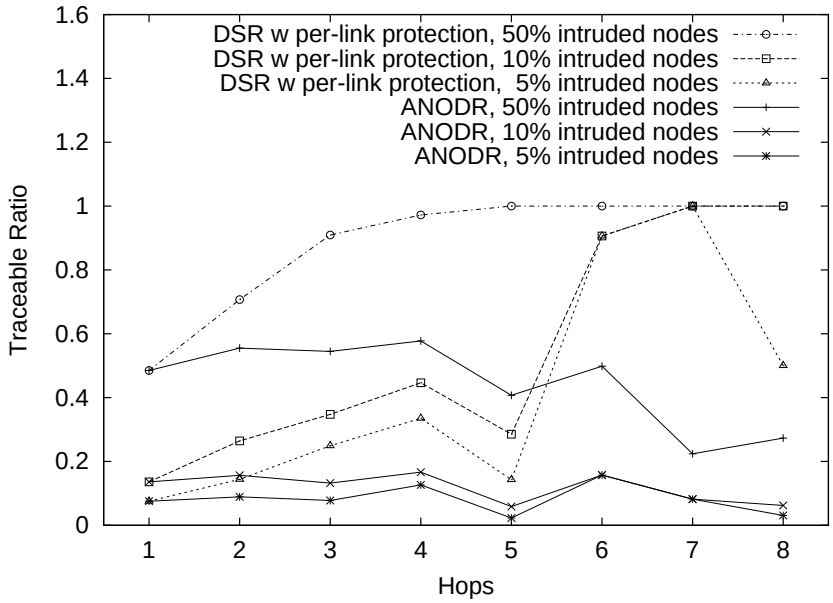

Figure 6.8: **Comparison of traceable ratios**

In the simulation a percentage of network members are marked as intruded. Figure 6.8 depicts the traceable ratio over different path lengths of routes for ANODR and DSR. Simulation uses 100 random CBR pairs each generating only one packet and nodes move in 2 m/s. The following table gives the path length distribution over all the connections. The results are averaged over 4 runs with different seeds.

hops	1	2	3	4	5	6	7	8
# of routes	45.25	19.5	20.25	6.75	4.25	3	0.5	0.5

The figure shows that starting from paths of only one-hop, where the two protocols expose the same amount of information (approximately same as the percentage of intruded nodes), the two protocols diverge into different trends. For DSR, traceable ratio increases when path length increases, due to the fact that longer paths are more likely to have intruded forwarding nodes. As a result, having as low as only 5 percent of intruded nodes, DSR's traceable ratio will be larger than 20 percent for paths longer than 2 hops. With 50 percent intruded nodes, DSR's traceable ratio quickly approaches 100 percent (reaches 90 percent at 3-hops long paths) when path length increases. In the graph, we see special cases in paths of 7 or more hops. This is because the chance of constructing long paths is rare in DSR simulation scenario. Even with multiple runs, the occurrence is too rare for meaningful statistics.

In contrast, ANODR is not sensitive to path length because the knowledge exposed to intruders is localized. Figure 6.8 shows that in general the traceable ratio of ANODR stays at the percentage of intruded nodes. When path grows longer, the traceable ratio will not exceed the percentage of intruded nodes. The result demonstrates ANODR's resistance to strong adversaries with node intrusion capability.

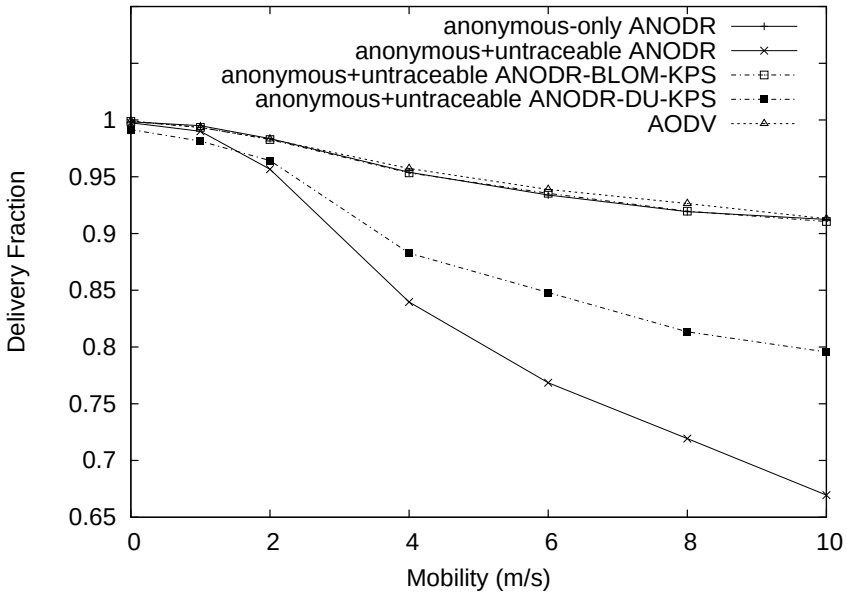

Figure 6.9: **Data Packet Delivery Fraction**

6.5.2.2 Routing Performance

Figure 6.9 gives the packet delivery fraction as a function of increasing mobility. The figure shows that "anonymous +only ANODR" and "anonymous +untraceable ANODR-BLOM-KPS" perform almost as good as optimized AODV. This result can be justified by the following three reasons: (i) The onion and/or the G-matrix-column exchange information used in ANODR control packets and the route pseudonym field used in data packets are not big enough to incur noticeable impact to the packet delivery fraction. (ii) The 0.02ms/1ms cryptographic computation overhead for the two schemes is too small to make a difference in route discovery. The latter reason also explains why the performance of "anonymous+untraceable ANODR" degrades faster than "anonymous-only ANODR" – their long computation time prolongs the route acquisition delay, which reduces the accuracy of the newly discovered route, leading to more packet losses. (iii) The route optimization of AODV has less effect when a network is at a medium size - 150 nodes (recall that our simulations for 50 nodes show

slightly higher delivery ratio of AODV). Further, the figure shows that "anonymous +untraceable ANODR-DU-KPS" has lower delivery ratio than "anonymous +untraceable ANODR-BLOM-KPS". The reason for the degradation is the failed probabilistic key agreement along the RtREP path. The source only has 0.9^k (k is the path length) chances of receiving a RtREP, which is a small number for a 150 node network. The source has to initiate a new route discovery in the absence of an expected RtREP, resulting in higher control overhead and lower performance. Clearly, the figure shows the tradeoff concern between the performance and the degree of protection. Fortunately, even with a much stronger protection provided by "anonymous+untraceable ANODR-DU-KPS", performance only degrades to 10 percent less than optimized AODV.

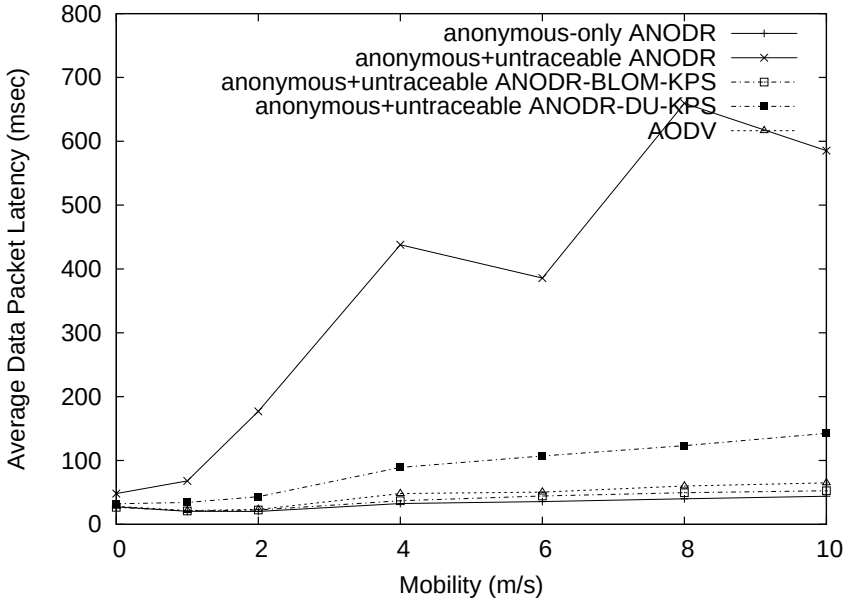

Figure 6.10: **End-to-end Data Packet Latency**

Figure 6.10 shows the average end-to-end data packet latency when mobility increases. "anonymous-only ANODR", "anonymous+untraceable ANODR-BLOM-KPS" and AODV exhibit very close end-to-end packet latency as they require very small processing time. "anonymous+untraceable ANODR" has much longer latency than

the aforementioned three due to additional public key processing delay during RtREP phase. “anonymous+untraceable ANODR-DU-KPS” also has longer end-to-end packet delay than the three due to the failure of RtREP packets and new route discovery. The delay trend of all the protocols increases when mobility increases, since the increasing mobility increases packet loss which triggers more route discovery, leading to increasing buffering time in waiting for a new route.

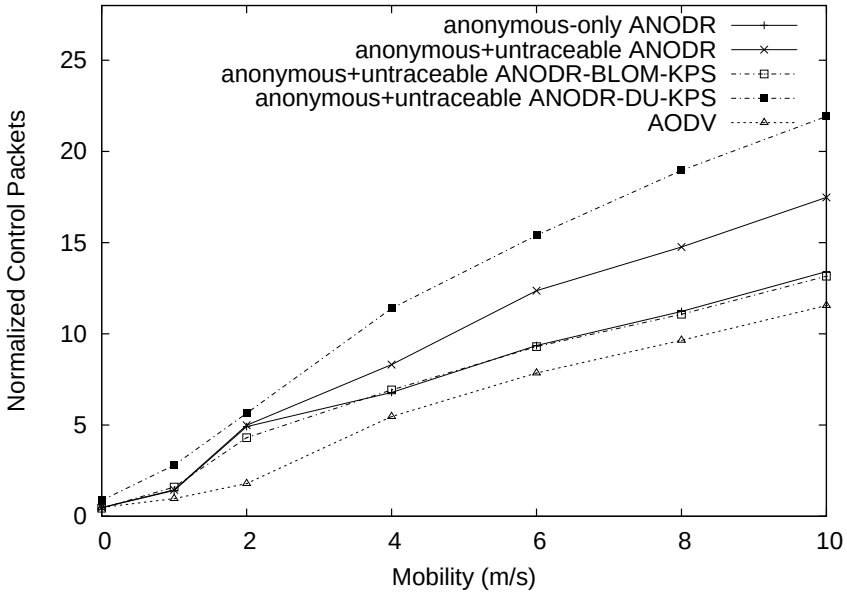

Figure 6.11: **Normalized Control Packets**

Figure 6.11 shows the number of transmitted routing control packets for each successfully delivered data packet. The figure shows that when mobility increases, all the protocols generate more control packets due to the fact that more links break at higher mobility, leading to more route discoveries. However, all the ANODRs generate larger number of control packets than AODV at each mobility point. The reason is that without routing operation optimizations, the four protocols rely more on route discoveries from the sources for repairing broken links. Especially, “anonymous+untraceable ANODR-DU-KPS” incurs more route discoveries than the others due the high loss

possibility of RtREP packets. “Anonymous+untraceable ANODR”’s long processing overhead on RtREP packets also causes more timeouts at the sources, triggers more route discoveries and generates more control packets.

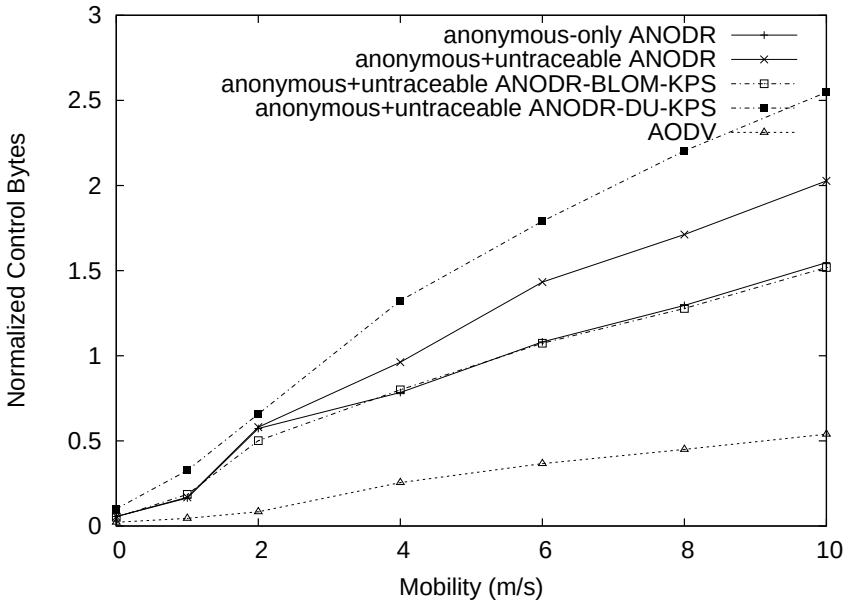

Figure 6.12: **Normalized Control Bytes**

Figure 6.12 gives the number of control bytes being sent in order to deliver a single data byte. The figure shows that all the ANODR variants send more control bytes than AODV. This result is expected, because they use larger packets due to global trapdoor, padded cryptographic onion and key. The comparison order of the curves are the same as indicated in Figure Figure 6.11 as expected and for the same reasons. When mobility increases, the figure shows the normalized control overhead grows in all the schemes as more control packets are transmitted for path recovery. The lack of optimization in ANODR variants demonstrates here a faster increasing trend as more recovery are generated from sources so more control overhead is produced.

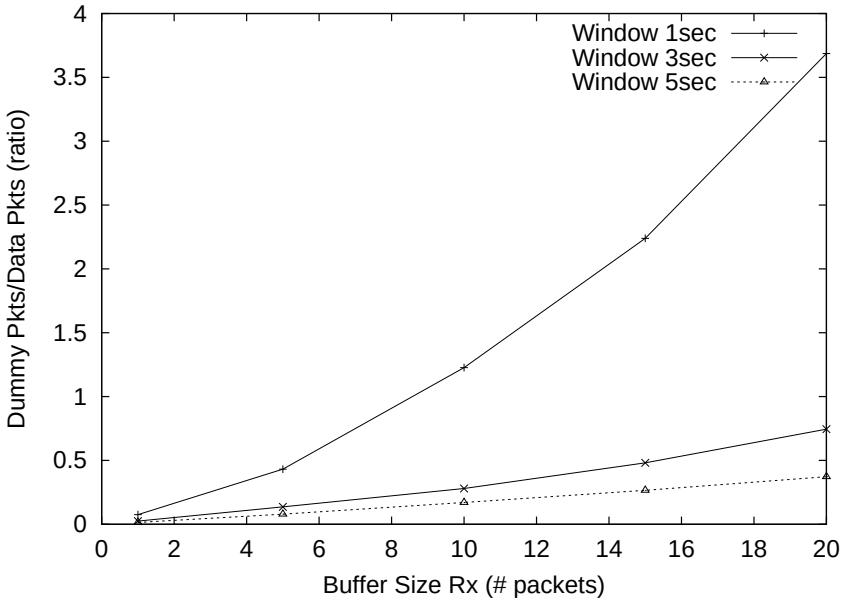

Figure 6.13: **Overhead with Mixing**

6.5.2.3 Neighborhood mixing performance

Figure 6.13 shows the ratio of dummy packets transmitted over actual date packets transmitted. It suggests that for a fixed playout time window size t_X, the larger the playout buffer size r_X is, the more dummy packets need to be transmitted according to the formula $r_X - r$. The figure also shows that when the playout time window size t_X increases, less dummy packets are transmitted due to the increment of value r accumulated over the time window. In many cases, the dummy packet ratios are reasonably small (say, less than 100% such that averagely at least one of two transmitted data packets is real). This demonstrates that mixing technique is practical in mobile ad hoc networks if appropriate values of playout window size and buffer size are selected.

However, it is a non-trivial problem to choose the best values for playout window size t_X and buffer size r_X. Many ad hoc network dynamics, including distributed decision making, wireless bandwidth estimation, end-to-end application latency requirement, and pre-defined lower bound metrics for t_X and r_X, have significant impacts on

the choice. It is appealing to employ an adaptive scheme to replace the fixed scenarios simulated in this work.

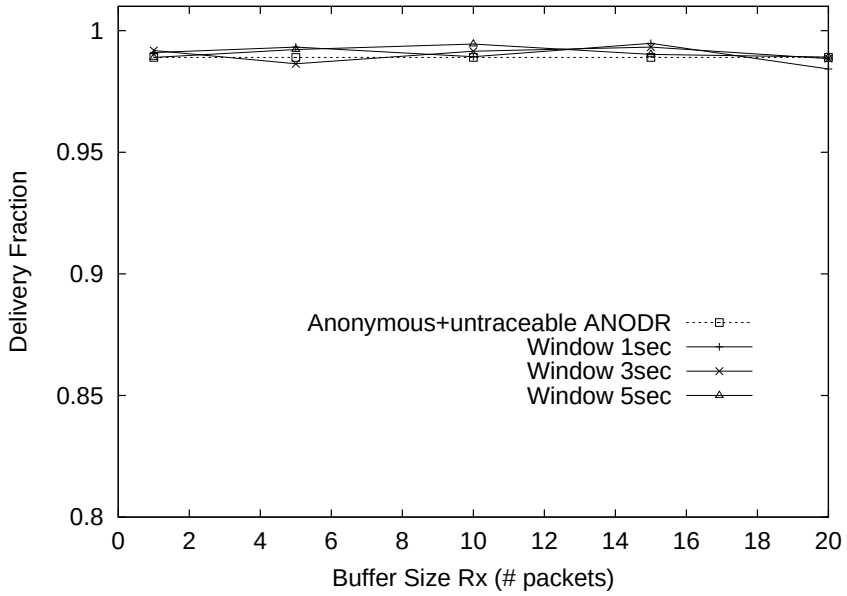

Figure 6.14: **Data Delivery with Mixing**

Figure 6.14 shows the packet delivery faction under the same mixing conditions as used in Figure 6.13. As a comparison, "anonymous+untraceable ANODR", which has been extensively studies in previous subsection, is presented here. The mobility parameter used in this experiment is equal to 1. The figure shows that "anonymous+untraceable ANODR" and its mixing variants perform closely. Some randomness occurs in the figure, but it does not suggest noticeable performance degradation. Thus the result suggests that the mixing packets generated under the current conditions do not affect the data packet delivery much.

6.6 Summary

In this chapter we proposed (standard) TIMBA model and ANODR routing protocol to ensure mobile anonymity for mobile nodes. The TIMBA model relies on two security policies, namely time interval policy and broadcast policy, to ensure perfect mobile anonymity in a dynamic network of peer nodes. The protection covers all mobile anonymity aspects, namely *sender/recipient identity anonymity*, *sender-recipient identity relationship anonymity*, *sender/recipient venue anonymity*, *sender-recipient venue relationship anonymity*, *location privacy*, and *motion pattern privacy*.

Nevertheless, TIMBA is an impractical ideal model. Due to performance concerns, ANODR makes compromises with the real world. ANODR does not enforce time interval policy in its REQ control traffic, and replace the policy with neighborhood traffic mixing in other traffic. It is possible to use timing analysis to break *sender venue anonymity* in ANODR. Nevertheless, ANODR extensively explores two readily available broadcast mechanisms, namely multi-hop on-demand flooding and single-hop omnidirectional wireless transmission, to enforce other mobile anonymity aspects. Among three ANODR variants, (1) the performance of "anonymous-only ANODR" is comparable to common ad hoc routing schemes, but it only ensures *sender/recipient identity anonymity*, *sender-recipient identity relationship anonymity*, *weak location privacy*, and *weak motion pattern privacy*; (2) "Anonymous+untraceable ANODR-KPS" features similar routing performance, and it additionally protects *recipient venue anonymity* and *sender-recipient venue relationship anonymity*; (3) "Anonymous+untraceable ANODR" additionally protects *strong location privacy* and *strong motion pattern privacy*. However, its routing performance is the worst, though better than public key intensive MIX-Net.

CHAPTER 7

Anonymous Routing Resilient to Active Attacks

It is not enough to succeed.
Others must fail.
–Gore Vidal

In this section we study how to ensure routing privacy and routing integrity in a single solution. We use data forwarding service as an example to demonstrate the usefulness of partial trust community and community-based communication. In addition, we will study how to provide basic cryptographic service, namely community-based key management, on an ad hoc route so that per-hop encryption and authentication are feasible.

7.1 Ideal models against active attacks

As far as mobile anonymity is concerned, the ideal TIBA model is unaffected by introducing active adversary into the system. An active adversary may disrupt a transmission event, but cannot compromise the perfect mobile anonymity ensured by TIBA.

However, in TIMBA model an active adversary can inflict more damage than traffic disruption. In particular, people have demonstrated several active attack strategies to break anonymity protection. Let's use tagging attack [34] as an example. In tagging attack, an active internal adversary en route modifies a message by embedding a

pattern (for example by flipping a bit) so that the pattern can be recognized later by another internal adversary en route. Like this example, other active attacks also seek to break anonymity protection by modifying the default protocol behavior. Some researchers believe it is nearly impossible to address all active attacks with justifiable expense [15]. It is easy to verify the belief by some counterexamples: In TIMBA, if active internal adversaries can form a cut[1] in the hypercube network, so they can drop all REQ/REP/DATA/ERR traffic across the cut, then they can definitely know whether a sender and its recipient are in the same partition or not. There is no way we can achieve perfect mobile anonymity against such attackers.

But some active attacks can be countered. In TIMBA, a dishonest adversary can sends back an REP packet even if it is not the real recipient. This can be countered by asking for anonymous proof of global trapdoor opening. In ANODR the design of anonymous proof pr_{dest} uses cryptographic commitment — the real recipient needs to present de-committment information in REP packet. We can trivially realize this design in standard TIMBA by adding a commitment field in (REQ) global trapdoor field, and a de-commitment field in REP packet.

	encrypted with ← recipient's key →		set local trapdoor ← per-hop →	for per-hop ← key agreement →
REQ (2 bits)	global trapdoor	commitment	onion	one-time public key
	(fixed length)		(fixed length)	(fixed length)

	← encrypted per-hop using the recorded temporary public key $public_{upstream}$ →	
REP (2 bits)	decommitment	onion
	(fixed length in a specific public key cryptosystem, e.g., 1024 bits)	

Message injection and message modification are common active attacks. Modification means some bits are different from what the source sent out, and injection

[1] So far we assume per link bandwidth is infinite. If per link bandwidth is finite, such cut can be formed by either a node-cut or a link-cut caused by distributed denial-of-service attack [80].

means some bits not from the source are injected. By their definitions the countermeasures are source-centric. Adding integrity support in each REP/DATA/ERR packet is obviously a feasible solution[2]. To prevent untrustworthy forwarders from modifying or injecting REP/DATA/ERR packets, a packet originator needs to share a key with each forwarder, and prepare a list of integrity checksums per forwarder per packet, so that each forwarder can independently verify packet integrity. Since the longest path is $(2 \cdot \log_2 N - 1)$ hop long, this design requires adding a data field of $(2 \cdot \log_2 N - 1)$ slots to each (REP/ERR/DATA) packet. Plus the public key signature overhead (or symmetric key agreement overhead if HMAC is used), this design incurs tremendous computational and communication overheads. This is still not enough — MIX-net and TIMBA is a perfect place for chosen ciphertext attack (CCA1) and adaptive chosen ciphertext attack (CCA2) because each MIX or TIMBA node is a decryption oracle of the onion. Therefore, we also need non-malleability support [42] which is equivalent to security against CCA [12]. It is even more challenging to secure REQ packets because during REQ phase no cryptographic protection is feasible. A trivial answer is no answer — we simply ignore active attacks against REQ flooding, and hope the destination will select the right REQ to reply just by probabilistic chance.

In previous anonymity research efforts, zero knowledge (ZK) proofs and threshold secret sharing are used to stop active internal adversary. By ZK proof, it is possible for each network member to prove that a set of its input elements actually correspond to a set of its output elements, while there is no (non-negligible) information leaking to other parties including those internal and external adversaries. Either interactive or non-interactive ZK proof is used in previous research efforts to ensure publicly or universally verifiable mixing [100][1][123][102][35][70] [71][94]. By threshold secret sharing, a quorum of network members cooperate to verify a message, while

[2] In TIMBA we can trivially add per-hop HMAC checksum to every data packet using the per-hop symmetric key. But this can only defend active external adversary.

the overall system can tolerate a threshold number of internal adversaries. Verifiable secret sharing [31] [51] [27] and publicly verifiable secret sharing [135] [128] use ZK "witness" information to detect incorrect shares injected by internal adversary. Recently, there are more relaxed enforcement mechanisms proposed to counter active attacks using very efficient operations, as studied in partial checking [72], almost-entirely-correct mixing [22], and reputation-based mixing [41][39].

Among these proposals, we believe non-malleable NIZK (Non-interactive Zero Knowledge) proof [122][124] is the best theoretic answer to the problem. NIZK plays a central role in building provably secure public-key cryptosystems based on general complexity-theoretic assumptions that achieve security against CCAs. In TIMBA, we can let the sender prepare a NIZK proof per packet per forwarder, then enforce an anti-disruption policy like this: any node can publicly detect packet modification if the NIZK proof prepared for a forwarder cannot prove that the packet being forwarded by the forwarder is valid. This realizes an anonymous and ubiquitous intrusion detection system that can detect disruption attacks anywhere anytime. The notion of NIZK is introduced by [18]. NIZK based on a single random string is a realistic NIZK model between a prover and a verifier sharing a common random string. In [85][49][47][48], it was shown that such realistic NIZK proofs exist for any language in NP assuming the existence of trapdoor one-way permutations.

Adding $(2 \cdot \log_2 N - 1)$ NIZK fields per TIMBA packet is easy. But such design is impractical because it imposes a heavy computational and communication overhead. In this chapter we will adopt a more practical approach — we explore the community-based routing approach to mitigate active attacks. The quest for the design is presented below.

7.2 The concept of "partial-trust community"

Internal adversary presents a great challenge to network security schemes. We trust a network member because we assume that the node will always keep its cryptographic keys as secrets. Once this assumption is invalidated by the internal adversary model, the employed cryptosystem provides no protection to a gullible network. In the presence of internal adversary, it is clear that *partial trust* is a fundamental problem in self-organized ad hoc networks. Each ad hoc node must make its decisions on whether to trust its ad hoc peers, and how to regulate the conferred trust.

Partial trust requires that a network service to be securely distributed to a community. We will call such a community as a "*partial trust community*" throughout the work. At the level of each individual node, the service provisioning is untrustworthy and is allowed to be disrupted. However, at the level of community, the service provisioning becomes trustworthy—even if some of community members are selfish or malicious, the service remains available and reliable. The challenge of realizing such a trustworthy community lies in three aspects:

- *Community creation and configuration*: A community can be created and configured anywhere and anytime, but the related process should only incur reasonably low cost.

- *Community maintenance*: The community must adapt to changes in the network topology and other dynamics. The impact of mobility, channel dynamics, community member join and leave, and the presence of malicious nodes must be addressed appropriately.

- *Application demands*: Depending on a specific application context, the community creation and maintenance process must also satisfy extra constraints. In other words, a community design is application dependent. For example, if

we want to provide location privacy and anonymity service to ad hoc nodes, it is not appropriate to implement community creation via a neighborhood discovery protocol. This is because any internal adversary can always abuse such neighborhood discovery protocol to correlate its neighbors with its location, thus compromises location privacy.

In a nutshell, when internal adversary is feasible, we believe an appropriate adversary model should be a new one that is very different from a cryptographic adversary. This adversary can compromise network services provided on an individual node, but cannot compromise network services provided by a well-configured and well-maintained partial trust community.

7.3 Network assumptions

At network layer, community-based communication is applicable to general routing schemes. Following the main trend of current research, we assume that the underlying routing protocol is an on-demand routing scheme. While proactive routing protocols exchange routing information even when there is no data transmission, the on demand approach pays the cost of routing overhead only when it is needed. An on-demand routing protocol is composed of two parts: *route discovery* and *route maintenance*. In route discovery, the source sends out a route request (RtREQ) to the network when it needs a route towards the destination. Other network members either forwards the RtREQ if it does not know any route to the destination, or otherwise sends back needed routing information to the source. Upon receiving route requests, the destination sends back at least one route reply (RtREP) to the source. Contrary to RtREQ flooding, an RtREP message is typically forwarded by a limited set of chosen forwarders, which are called "*RtREP forwarders*" (or "*RtREP nodes*") in this paper. Although various

on-demand routing protocols use different algorithms to process RtREQ and RtREP messages, the combination of RtREQ and RtREP processing establishes a route between the source and the destination. Due to mobility and network dynamics, an established route may be broken at any time. On-demand routing schemes use route error (RtERR) notification to inform the source or the destination about the status. Then the source will initiate a new route discovery procedure to find new routes towards the destination.

Given a general on-demand routing scheme (e.g., AODV [108], ARAN [125], DSR [73], Ariadne [65]), *the original RtREQ procedure is unchanged, and the chosen set of RtREP forwarding nodes is also unchanged*. In other words, RtREQ processing and RtREP nodes are unaffected by our design. By this assumption, it is possible to seamlessly integrate community-based communication with existing ad hoc routing schemes.

At link layer and below, we assume that a node can always monitor ongoing transmissions if the node itself is not transmitting. In addition, wireless links are symmetric; that is, if a node X is in transmission range of some node Y, then Y is in transmission range of X. Using 802.11 style MAC protocol, a node can broadcast to its neighbors using CSMA, or unicast to a specific neighbor using CSMA/CA. In community-based communication, we explore node redundancy in a self-organized dense network to stop disruption attacks. We assume that in a network locality there are usually some redundant network members. These peer members will have identical capabilities and responsibilities in community-based communication. No centralized control or hierarchical control is assumed.

7.4 Design principles

All-or-nothing disruption By launching rushing attack [67], wormhole attack [66], or simply due to probabilistic chances, it is possible that an internal adversary is selected into the forwarder set of a connection. Our design may not completely stop all MANET security attacks. But we seek to enforce ineluctable constraints to regulate the internal adversary's behavior. In the ideal case, the adversary has to choose between two choices. (1) "**All**": The adversary has to completely bring down the connection and to obliterate the current route; (2) "**Nothing**": The adversary tolerates community-based communication and treats the balanced result as an acceptable answer. In the former case, since on-demand routing schemes follow a soft-state design, the source will finally reach a timeout and launch another route discovery procedure to find new routes. Hence the adversary is relieved from its control over current route. In the latter case, route disruption is mitigated to a level acceptable to both the legitimate side and the adversarial side.

End-to-end maintenance Due to the presence of internal adversary, the intermediate forwarders are not fully trustworthy. Therefore, the two ends of a connection should pay reasonable cost to maintain good connection status. End-to-end maintenance may include monitoring end-to-end data delivery ratio and latency, implementing end-to-end polling, maintaining fresh route status, and finding new routes when current route is broken or compromised.

Severe environment assumption Untethered ad hoc nodes may be deployed in hostile environments to serve mission critical applications. The adversary may explore *all* practical means to attack the ad hoc network. These attacks include not only active disruptions [65][125], but also passive traffic analysis [78] such that the adversary can find the most efficient way to disable our ad hoc network. We believe that a secure

routing protocol designed for mission critical applications should address both active and passive attacks. The needed routing security support is comprised of two parts: routing integrity and routing privacy. The former part mitigates route disruptions, and the latter part ensures anonymity and location privacy for mobile nodes.

Intrusion tolerant design An important observation made in this work is that popular intrusion detection design and popular neighborhood discovery design are both incompatible with the combination of routing privacy demands and internal adversary. In the presence of internal adversary, intrusion detection directly conflicts with anonymity and location privacy requirements. An intrusion detection system requires all network members to identify themselves so that a centralized or distributed detection algorithm can authenticate and monitor suspected nodes [147]. Similarly, a neighborhood discovery design requires all mobile nodes to reveal their identities to their neighbors. Unfortunately, any *internal* adversary can always explore this chance to compromise location privacy in its neighborhood. As location privacy is considered as one of the application demands, we would exclude neighborhood discovery schemes and adopt an intrusion tolerant approach in answering the challenge imposed by internal adversary.

7.5 Community-based communication

Establish and maintain partial trust community is the central part of our community-based communication scheme. For each communication session, a partial trust community is established at each forwarding step to thwart route disruption. This section details how a community at each forwarding step is created and how the communities are maintained facing network dynamics and possible attacks targeting the communities.

7.5.1 Community configuration

The community concept is based on such an observation that multi-hop wireless forwarding always rely on nodes in the current radio transmission range to relay packets. Figure 7.1 shows the simplest case that node B relays packets from node A to node C. Typically, node B is within the intersection of node A and C's radio range while A and C cannot hear each other. In principle, all nodes within the intersection can reply packets from A to C. Nodes in such a intersection forms our *community*. Figure 7.2 depicts a chain of communities along a multi-hop path. The community-based communication is to explore the node redundancy within the community at each stop so that the conventional per-node based forwarding scheme is seamlessly converted to a new per-community based forwarding scheme. Intuitively, this simple design can be viewed in a straight-forward way—an partial trust community is a "big virtual node" that replaces a single forwarding node in conventional routing schemes (Figure 7.3).

Based on general on-demand routing mechanism, a simple design option is to construct such a community between any pair of RtREP nodes that are two hops away, e.g., Figure 7.1 shows a community around RtREP node B that is built between node A and C. There are other relatively more complex options. For example, one can use a variant of neighborhood discovery protocol to identify local nodes and to evaluate nearby traffic metrics. Then local communities can be formed by selecting appropriate local nodes based on traffic evaluation. As we argued previously, neighborhood discovery protocols are excluded in our design because any internal adversary can always use such protocols to break location privacy in its neighborhood. We adopt simple approaches due to the concerns of controlling design complexity and providing location privacy services.

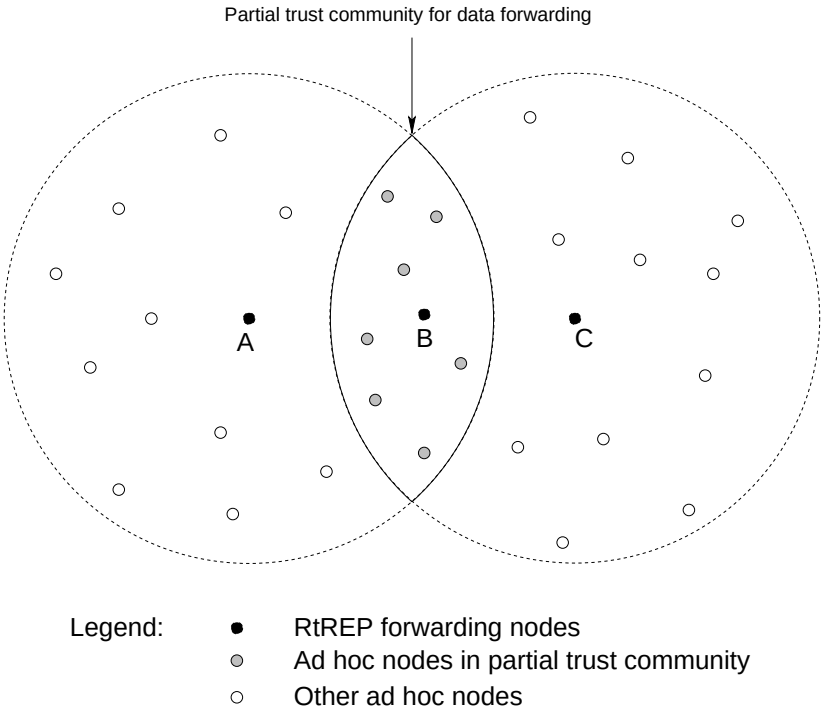

Figure 7.1: **A data forwarding partial trust community between a 2-hop source and destination pair**

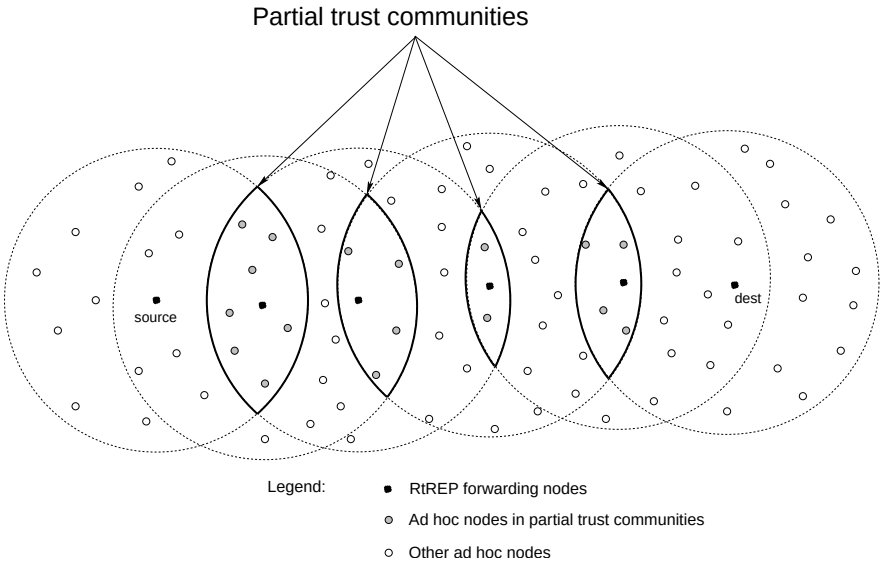

Figure 7.2: **Data forwarding partial trust communities along a multi-hop path**

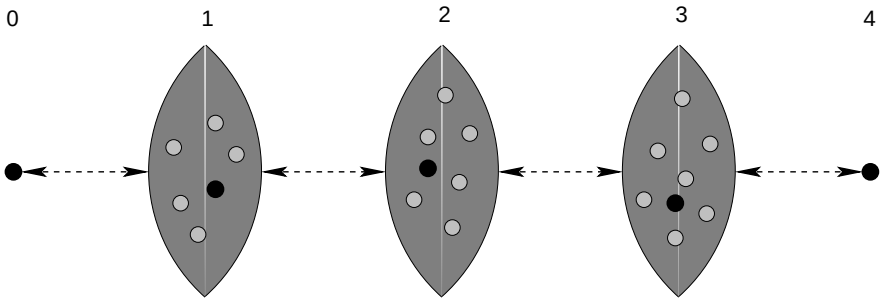

Figure 7.3: **Partial trust communities as “big” virtual nodes**

7.5.2 Community creation and maintenance

Due to mobility and other network dynamics, the membership of an partial trust community is not static. Each node must maintain a membership flag per connection/route. The flag is a soft-state one, which expires on a timeout T_{expire}. For each connection, periodical polling between the source and the destination is required.

- The polling interval T_{poll} is used to determine an appropriate T_{expire}. Given dynamically measured connection status, it is feasible to adapt T_{poll}. An appropriate T_{poll} must be larger than the estimated round trip time RTT between the source and the destination. That is, $T_{poll} > RTT$, and we suggest $T_{expire} = 2 \cdot T_{poll}$.
- End-to-end periodical polling helps maintaining communities en route. The polling messages and associated replies can be used by ad hoc nodes to determine its community membership. A simple design option is to let each ad hoc node set the membership flag upon overhearing at least three consecutive polling messages (or three consecutive polling replies, whichever choice is better). This is because a community member must be in the transmission range of at least three forwarders: the immediate upstream forwarder, the forwarder in the same community, and the immediate downstream forwarder.
- End-to-end polling helps the two ends (the source and the destination) to monitor untrustworthy intermediate forwarders. An end-to-end polling scheme was recently proposed by Awerbuch et al. [8] to detect internal adversaries. Variants of the scheme can be devised to meet other security demands.

To correctly maintain the communities immediately next to the source/destination, the source/destination needs to send one more polling message (or reply) to itself. Besides, due to wireless channel contentions and errors, it is possible that a *de facto*

community member fails to overhear at least one of the three messages/replies. Fortunately, this unlucky node has the chance to rectify its incorrect membership flag at the time of next polling. Third, it is possible that a node hears more than three consecutive polling messages. This is because RREP nodes may be near each other due to mobility and other reasons. Then nodes may be shared by more than one partial trust communities. Such nodes can autonomously choose one of the several overlapping communities as its community.

In addition, for performance concerns, the polling messages and replies must be short packets or piggybacked short data fields. Combined with a reasonable polling interval T_{poll}, the communication overhead incurred by polling is tractable.

7.5.3 Data forwarding

Community-based data forwarding is a combination of conventional node-based data forwarding plus a community backup mechanism. At the source, the source node is unambiguously the current forwarder. At each intermediate stop, the RREP node is supposed to be the current forwarder. This RREP node plays the role of "*core*" in the associated partial trust community. However, if this node fails to forward data packet due to maliciousness or selfishness, members in the associated partial trust community will forward the data packet based on probabilistic competition—in 802.11 MAC the first node who wins the chance of transmission after randomized exponential backoff is the backup forwarder.

Data is delivered by wireless unicast from current forwarder to the next core. This unicast is overheard (and decrypted if the unicast is protected by per-hop encryption) by all nodes in both communities. If any anomaly is found, members in the current community will try to forward the data packet again. If data forwarding has repetitively failed for a threshold number of time, at least one RERR notification is sent towards

the source or the destination. The route is hence broken and "all-or-nothing disruption" is achieved.

7.5.4 Attacks against community maintenance

Any newly introduced design will be attacked once it is realized. Therefore, the security aspects of any new design must be analyzed and re-evaluated upon every chance. This observation is applicable to community based communication with no exception.

First, an adversarial forwarder can drop polling messages or replies. However, since the source cannot receive the polling reply within T_{expire}, it will initiate route discovery procedure and find another route. This lessens the adversarial node's chance to disrupt ad hoc routing.

Second, an adversarial forwarder can disrupt community maintenance by modifying valid polling messages/replies, or injecting fake messages/replies. In the latter case many non-community nodes will believe they are in a community and erroneously set their membership flags. To address these attacks, polling messages and replies must be appropriately protected such that modification and injection are infeasible.

Here we propose a countermeasure to defend these attacks against community configuration and maintenance. The essential defense mechanism is to implement damage control along a multi-hop path, such that adversarial forwarders cannot compromise those wireless links under the control of "good" nodes. By this design choice, each peer node claims its responsibility for its own link, and adversarial nodes cannot claim responsibility for links that are out of its neighborhood. This realizes cryptographic partial trust.

Cryptographic onion An important property of this onion design is that each forwarder's capability is strictly restrained in the corresponding layer created for it. Each

forwarder cannot interfere with other layers unless it can break cryptographic encryption. Similar onion data structure was firstly used in Byzantine node (internal adversary) detection protocol proposed by Awerbuch et al. [8]. Later we also used the same data structure in their anonymous on-demand routing protocol [78]. Based on these previous efforts, we observe that cryptographic onion is applicable to ad hoc network security due to following reasons:

- *Ordered traversal and conditional loop-free*: The onion is formed in a strict order and protected by cryptographic means. No intermediate forwarder is able to change the order unless it can break cryptographic protection. Besides, if there is no routing loop during route discovery (i.e., RREP successfully goes back to the source), then the ad hoc route enforced by onion traversal is loop-free.

- *All-or-nothing modification*: The layered cryptographic design prevents a forwarder from attacking other nodes by removing specific nodes from the onion. A malicious forwarder may drop the entire onion, then the source cannot receive polling reply within T_{expire} and will initiate another route discovery to find new routes. Hence this attacking strategy will also relieve the adversary from its control over the current route. An equivalent form of dropping an onion is to pass the onion to next stop without decryption at current stop. Then the next stop has to drop the onion because it cannot decrypt it. Nevertheless, this does not change the effect of "all-or-nothing disruption".

- *Distributed processing*: The processing of the onion is distributed to each peer node. Each peer node's autonomous decision is equally respected in the design. This property is compatible with a self-organized network formed by peer ad hoc nodes.

The onion used in polling is formed during route discovery phase. The source

piggybacks a 128-bit random data in its RREQ message. This random block of data serves as the "kernel" of an onion. Each RREQ forwarder adds a layer of onion by (1) selecting a personal random key, (2) storing the key under a record reserved for the corresponding connection, and (3) encrypting the current onion using the chosen key. When this onion reaches the destination, it can be used by the destination to implement a reversed traversal along the RREQ forwarding path.

Similarly, the destination can piggyback a 128-bit random data in its RREP message. Each RREP forwarder adds an onion layer in a similar way. When this onion reaches the source, it can be used by the source to enforce ordered packet forwarding towards the destination.

Polling design details Polling messages and replies are either short packets or short data fields piggybacked on data packets. At this time the source/destination already caches the onion obtained during route discovery phase. This onion is part of a polling message, which is in the following format:

$$\langle POLL, seq\#, onion, hop\_count \rangle.$$

Here $seq\#$ uniquely identifies each connection (e.g., the same source $seq\#$ used in RREQ). The hop_count field is a counter that is reset to 0 at the polling site, and is increased by 1 at each stop. Similarly, a polling reply is in the format:

$$\langle POLL\_REP, seq\#, onion, hop\_count \rangle$$

where hop_count is a counter reset to 0 at the polling reply site. The hop_count field is needed to identify neighboring communities. In particular, using the simple community maintenance algorithm described in Section 7.5.2, a community member is able to know which community it is in. For example, if a mobile node overhears three POLL

messages with hop_count values 2, 3, and 4 in the strict order specified, then it can conclude it is in the community indexed by 3.

7.6 More protocol attacks and countermeasures

The scheme described above is the basic version of community based communication. Depending on the strength of adversary, the basic version may not be able to counter some strong attackers. In this section we discuss these ad hoc routing attacks and their impact on community based communication. These attacks demand various improvements to the basic version.

7.6.1 Repeater attack

An active attack against our scheme is "*repeater attack*" [67], that is, an adversary forwards RREQ without changing the onion and/or the hop count. By this attack the adversary may let two nodes that are two-hop away believe they are one-hop neighbors. Then the adversary can go away and the ad hoc route is broken. This attack is addressed in the basic version because it is identical to dropping polling messages. Or if the adversary forwards polling messages, then a local community is formed as usual. Moreover, to mitigate this attack, each RREQ forwarder needs to monitor its neighborhood and RREQ packets with previously seen onion are ignored by an RREQ forwarder.

7.6.2 Wormhole and physical layer attacks

The major drawback of the basic version is that it assumes all network nodes, including adversarial nodes, use the same omnidirectional radio. This makes it vulnerable to physical layer attacks exploring non-omnidirectional radio transmission (e.g., wired

link or directional antenna) and variable-power omnidirectional radio transmission (e.g., software radio [134]). These attacks are closely related to "wormhole attack"[65] where two or more adversarial nodes have physical capability to shorten pairwise communication distance amongst themselves. Then the adversaries can violate network topological settings following their own will.

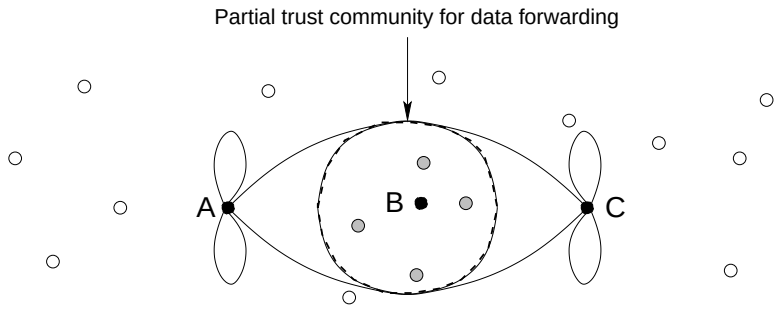

Figure 7.4: **Partial trust community using directional antenna** (assuming packet acknowledgement and every other adversarial RREP nodes)

- *Wired link*: Wired link requires extra hardware support on adversarial nodes. If two adversarial nodes at a single hop use wired link, then the basic version unconditionally fails.

- *Directional antenna*: Directional antenna requires extra hardware support on adversarial nodes. If two adversarial nodes at a single hop uses directional antenna, partial trust communities can be formed because the covered range of directional antenna happens to overlap with the community area in the basic version (as depicted in Figure 7.4). However, the adversaries significantly decrease the size of affected communities.

- *Variable-power omnidirectional radio*: On many off-the-shelf wireless interfaces, variable power management can be used with energy efficient forwarding to reduce energy consumption and enhance spatial reuse [75]. However, such

practice is a *de facto* attack that significantly decreases the size of affected communities.

The essence of all three physical attacks is to break the prerequisite of using omnidirectional radio with (nearly) identical transmission range. In this paper we suggest to use the following link layer countermeasure to enforce the prerequisite. We believe that better countermeasures can be devised given more physical layer supports, such as physical means to distinguish omnidirectional radio from directional radio.

Our link layer countermeasure requires that *one of the two RREP node at a single hop is not adversarial.* Here we call this constraint as the "*Colluding Constraint*". In the worst case, our countermeasure is effective when every other RREP node is adversarial. Given the fact that any non-adversarial "good" node will always use omnidirectional radio with a pre-defined transmission range, the problem is thus translated into enforcing symmetric communication between an adversarial node and a non-adversarial node.

RTS-CTS handshake can be used in 802.11 unicast to enforce symmetric communication. In particular, RREP packets and polling packets are transmitted in 802.11 unicast. Ad hoc security proposals like Passive Acknowledgements (PACK) [74] and "watchdog" protocol [92] also explore omnidirectional radio in detecting forwarding anomalies. They can be integrated into our link layer countermeasure to enforce symmetric communication.

In our worst case, an adversarial node is forwarding packets between two non-adversarial nodes. Since neither non-adversarial node will accept wired connection, the adversary has to use directional antenna and/or variable-power transmission with a lower-bound to reach both non-adversarial nodes. This means variable-power attacks is *not* effective if the *Colluding Constraint* is satisfied, and adversarial nodes must be equipped with extra directional antenna to decrease the size of affected communities.

Now we study the impact of the *Colluding Constraint* on our secure ad hoc routing scheme. Without loss of generality, let's suppose average hop count of an ad hoc connection is h and there are p $(0 \leq p \leq 1)$ internal adversarial network members. For each single hop on the h-hop connection, the probability that at least one node is *not* adversarial is $1 - p^2$. And the probability that the h-hop connection satisfies the *Colluding Constraint* is $(1 - p^2)^h$, which is depicted in Figure 7.5. Given the status quo of the scalability of current ad hoc routing protocols (DSR [74] currently suggests 5 or 10 to be the appropriate hop count), the probability is higher than 50% even when attacker percentage $p = 20\%$. Obviously one of our future work is to remove the *Colluding Constraint* in community based communication.

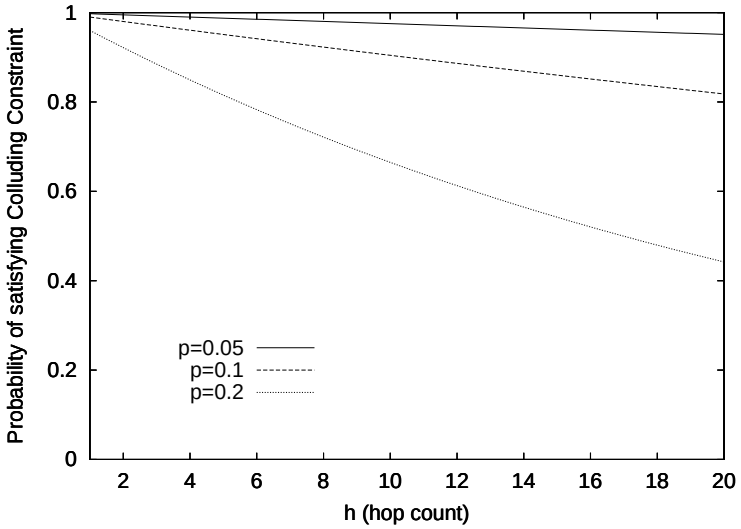

Figure 7.5: **Probability of satisfying the *Colluding Constraint***

7.6.3 Replay attack

Replay attack is not applicable to polling. If an adversary replays a polling message at a wrong place, the replayed message is ignored because its onion is out of sequence of neighboring onions. If an adversary replays a polling message at the right place

(i.e., in-between the two apposite non-adversarial RREP nodes), he is helping us to form the needed partial trust community. The essence of our onion design is to fix the forwarding relation along a multi-hop path formed during RREP phase so that no single node can change the forwarding relation along the path. The polling design itself is based on replay.

7.6.4 Enforce end-to-end communication

In the basic version each connection uses the same set of onions in its polling process—one polling onion from the source to the destination, and another reply onion on the reversed direction. An adversarial node can send back its reply onion right after it receives the polling onion. A possible countermeasure is to implement "one-time" onions (e.g., onions embedded with timestamp) proposed in [8]. But this requires the polling message originator to share a key with each forwarder. This requirement incurs non-trivial overhead on a constantly changing ad hoc route with variable number of forwarders. Instead, we employ a general cryptographic technique "trapdoor commitment" to enforce end-to-end communication in ad hoc networks.

The problem is stated as follows: Given two ends S and D of a connection, how does S ensure that a reply packet must be originated from D? The answer is that S adds a commitment field in its polling packets, for example,

$$K_{SD}(K_{reveal}), K_{reveal}(\alpha),$$

where $K(M)$ means using key K to encrypt message M, K_{SD} is the end-to-end key shared between S and D, α can be any well-known plaintext, and K_{reveal} is a random nonce selected for each polling. Then D presents a de-commitment field in its reply.

Using the previous example, the de-commitment field is simply

$$K_{decommit} = K_{reveal}.$$

Any forwarder of the polling reply can verify that

$$K_{reveal}(\alpha) \stackrel{?}{=} K_{decommit}(\alpha).$$

If it is a match, then the polling reply does originate from the destination D. Consequently, polling message and polling reply are changed into the following formats:

$$\langle POLL, seq\#, polling\_onion, commit, hop\_count \rangle,$$

$$\langle POLL\_REP, seq\#, reply\_onion, decommit, hop\_count \rangle.$$

7.7 A community-based key management protocol

In this section we realize cryptographic protections for community-based communication. We seek to establish a web-of-key in the network. In conventional node-based communication schemes, such web-of-key means that every pair of neighboring nodes share a random key. In community-based communication, the meaning of this requirement is changed to establishing a random key between two neighboring communities along a multi-hop route. This random key can then be used in cryptographic protocols to provide message privacy, message integrity, and other security services.

In a conventional routing scheme, it is possible to employ public key cryptosystems to establish pairwisely shared web-of-key. In other words, in a neighborhood any two nodes can independently exchange a secret key using schemes like Diffie-Hellman, RSA, or Elliptic Curves cryptosystems. Unfortunately, right now we have two neigh-

boring communities rather than two neighboring nodes. It would be too expensive to employ public key cryptosystem under this application context. Suppose community C_1 has $|C_1|$ members, and its neighboring community C_2 has $|C_2|$ members. If we do pairwise public key exchange (using algorithms like Diffie-Hellman, RSA and protocols like IKE), then there are $(|C_1| + |C_2|) \cdot (|C_1| + |C_2| - 1)$ possible combinations and each of them needs an individual key exchange. Even with optimization, for example, two "core" nodes do a public key exchange at first, then distribute the key to community members via public key exchange, we still need to do $(1 + |C_1| + |C_2|)$ public key exchanges. The incurred computational overhead is not apposite for mobile ad hoc nodes. In particular, Hu et al. [65] illustrated a feasible resource consumption attack against public key based protocols. An attacker can trivially pay little computational cost to flood a victim with packets containing random fake key exchange request, while the incurred crypto-operations can be prohibitively expensive for the victim. Contrary to this naive scheme, we describe below a probabilistic community key establishment algorithm for the two communities (of totally $w = |C_1| + |C_2|$ nodes). The algorithm only uses low-cost symmetric key crypto-operations and succeeds with high probability.

The community based key establishment is comprised of two parts: (1) a distributed key selection (DKS) protocol that was executed by off-line authority at network bootstrapping phase, and (2) a local key discovery (LKD) protocol to identify a common key at real time for two neighboring communities.

7.7.1 Distributed key selection

In a DKS protocol, each node is randomly assigned a specific number of keys from the universal set of keys to form its key ring. Here we will explore the concept of "cover-free family" (CFF) [45]. Most of the previous key distribution scheme [16] [21] [24]

[52] [83] [93] can be considered as a special case of the cover-free family scheme. The concept is formally defined as below [136].

Definition 5 *Let A be a set of N elements $\{a_1, a_2, \ldots, a_N\}$ and B be a set of subsets (called blocks) of A. Also let N and T denote $|A|$ and $|B|$, respectively. Then the set system (A, B) is called a $(w, r; d) - CFF(N, T)$* cover-free family *if, for any w blocks $X_1, \ldots, X_w \in B$ and any other r blocks $Y_1, \ldots, Y_r \in B$, we have*

$$\left|(\cap_{i=1}^{w} X_i) \setminus (\cup_{j=1}^{r} Y_j)\right| \ \geq \ d,$$

where d is a positive integer, and $\setminus$,$\cup$,$\cap$ denotes set difference, union, and intersection, respectively.

Intuitively, A is the universal set of keys (i.e., the key pool maintained by the off-line authority), $N = |A|$ is the key pool size, B is the set of nodes/users, $T = |B|$ is the number of nodes/users, and a block X_i or Y_j is a personal key ring.

The value w is the upperbound of number of key rings that are allowed to meet together and find at least d common keys. The value r is the upperbound of number of compromised key rings that cannot break the privacy of the d common keys. In other words, any subset of nodes with size up to w can find at least d common keys from their key rings, while any collusion of nodes outside this subset (with size smaller than r) do not have any knowledge about the common keys. Ideally, given a group of nodes/users of size T, our goal is to construct a $(T, r; d) - CFF$ in a fully distributed manner. When $d > 1$, some common keys can be revoked to allow higher tolerance for node compromise. Throughout this chapter we will simply construct a $(T, r; 1) - CFF$ for two neighboring partial trust communities C_i and C_{i+1}. The value $w = |C_i| + |C_{i+1}|$.

Recently Chan [25] has proven an important mathematic property of $(T, r; d) - CFF$:

By constructing a $(w, r; 1) - CFF$ over the universal key set A in a fully distributed manner, each node or user has identical responsibility and capability. A node or user can pick a block $X_i \subset A$ for his key ring. Based on the property of $(w, r; 1) - CFF$, any subset of nodes with size up to w can find at least one common key from their key rings and any collusion of nodes outside this subset (with size smaller than r) do not have any knowledge about the common key. Ideally, given a group of nodes of size T, our goal is to construct a $(T, r; 1) - CFF$ in a fully distributed manner. However, to allow higher tolerance for key revocation due to some compromised nodes, we usually construct a $(T, r; d) - CFF$ to accommodate revoked keys.

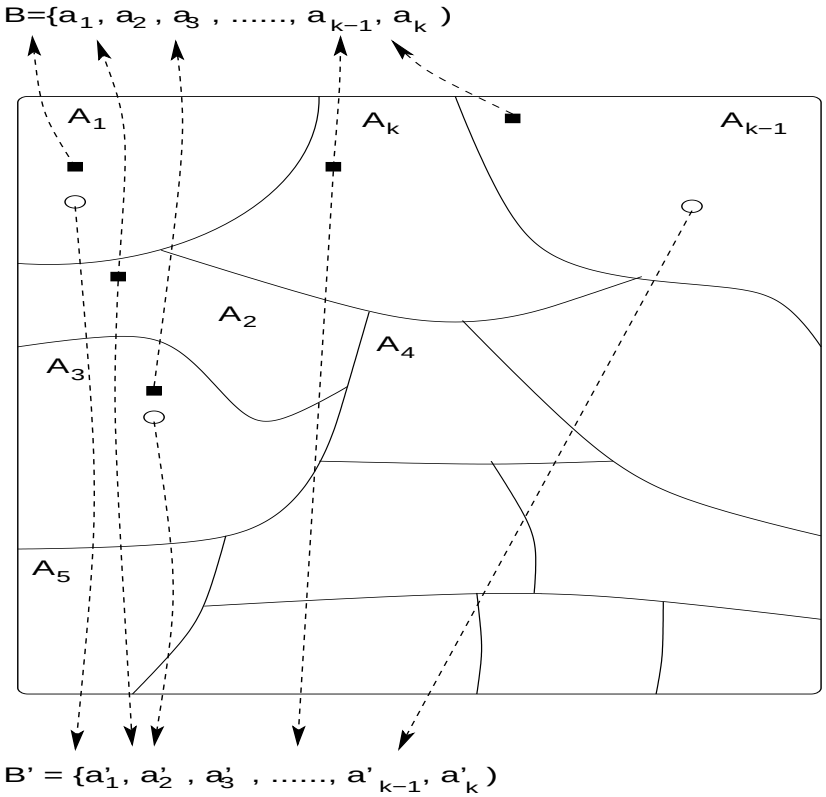

Figure 7.6: **Fully Distributed Key Selection (DKS) based on probabilistic CFF construction**

Figure 7.6 shows how two nodes can individually pick their keys in the probabilistic construction. In the figure A is known to the authority and k is publicly know. We also assume that the minimal capacity of a network node's key ring is k_B. The detail of a $(w, r; d) - CFF$ construction is as follows:

1. Select $k \leq k_B$ such that d divides k. (When $d = 1$, k can simply be k_B. k and d could be hard-coded.)

2. Form the universal key set A with size $N = kur$.

3. Divide A into k partitions $A_1, A_2, \ldots, A_k$ each of size ur.

4. Each node individually pick keys for his key ring to form $B = \{a_1, a_2, \ldots, a_k\}$ with each a_i randomly selected from the partition $A_i, 1 \leq i \leq k$. Each a_i is chosen following a uniform distribution over all elements in A_i.

Finally, this method yields a $(w, r; d) - CFF(N, T)$ with the following guarantee:

Theorem 5 ***(Chan's CFF Theorem)*** *The probability that this system is not $(w, r; d) - CFF(N, T)$ is at most e^{-t} if the following condition on T (the number of users) is satisfied:*

$$T \leq e^{\frac{2k \cdot \left(2 - \frac{d}{k} - \left(\frac{1}{ur}\right)^{w-1} \cdot e - \frac{1}{u}\right)^2 - t}{w+r-1}} \tag{7.1}$$

and $u = \frac{N}{kr}$.

PROOF: For a fixed block B_0, we select $w - 1$ other blocks $B_i, 1 \leq i \leq w - 1$ and r other blocks $C_j, 1 \leq j \leq r - 1$. For any $a_i \in B_0$, we have

$$p = Pr[a_i \neq \cap_{i=1}^{w-1} B_i \setminus \cup_{j=1}^{r} C_j]$$

$$
\begin{aligned}
&= 1 - \left(\frac{1}{ur}\right)^{w-1} \cdot \left(1 - \frac{1}{ur}\right)^{r} \\
&\geq 1 - \left(\frac{1}{ur}\right)^{w-1} \cdot e^{-\frac{1}{u}}
\end{aligned}
$$

Let $X_i = \delta[a_i \notin \cap_{i=1}^{w-1} B_i \setminus \cup_{j=1}^{r} C_j]$ where $\delta[\cdot]$ is the delta function, then

$Pr[X_i = 1] = p$, and $Pr[X_i = 0] = 1 - p$.

Let $X = \sum_{i=1}^{k} X_i$, and apply Chernoff bound to it, we have

$$
Pr[X > a] \;=\; Pr\left[\sum_{i=1}^{k} X_i > a\right] \;<\; e^{-\frac{2}{k}\cdot(kp+a)^2}
$$

Using the bounds obtained above, we have

$$
\begin{aligned}
Pr[|\cap_{i=1}^{w-1} B_i \setminus \cup_{j=1}^{r} C_j| < d] &= Pr[X > k - d] \\
&< e^{-\frac{2}{k}(k(1+p)-d)^2} \\
&= e^{-2k\left(1+p-\frac{d}{k}\right)^2} \\
&\leq e^{-2k\left(2-\frac{d}{k}-\left(\frac{1}{ur}\right)^{w-1}\cdot e^{-\frac{1}{u}}\right)^2}
\end{aligned}
$$

There are T possible choices for B_0 and $\binom{T-1}{w+r-1} \cdot \binom{w+r-1}{r}$ possible combinations for B_i and C_j leading to a total of

$$
T \cdot \binom{T-1}{w+r-1} \cdot \binom{w+r-1}{r} < T^{w+r-1}.
$$

The probability that the system is not $(w, r; d) - CFF$ is bounded above by

$$
T^{w+r-1} \cdot e^{-2k\left(2-\frac{d}{k}-\left(\frac{1}{ur}\right)^{w-1}\cdot e^{-\frac{1}{u}}\right)^2}
$$

Substituting the statement of the theorem, we can achieve an upper bound of e^{-t}. Q.E.D. □

It is clear that the allowed number of nodes/users in the network (T) increases exponentially as the size of key ring (k) increases linearly. That is to say, a key distribution scheme based on $(T, r; d) - CFF$ is potentially a scalable solution in terms of network size. In this work related security parameters are chosen as below:

- *Value* d: We choose $d = 1$ so that key agreement is feasible.
- *Value* t: The value t in Chan's Theorem is selected to be 2.3 such that the CFF failure probability e^{-t}=10%. Thus any w nodes can meet together and agree on a common key with 90% probability of success, and this key agreement cannot be discovered by any other r colluding nodes.
- *Value* w: w is approximately the sum of the size of two neighboring communities. This value depends on physical deployment of an ad hoc network. A typical value range is from 2 to 50. In this work we choose $w = 10$.
- *Value* r: We choose an r proportional to w. This means the number of internal adversaries that the entire network can tolerate overally is comparable to the community size. In particular, we choose $r = w$ in this work.
- *Value* k *and* T: In this work we choose $k = 50$. By Theorem 5, we know there must be a T (network size) satisfying Inequation (7.1) for $w = r = 10$, $d = 1$, $k = 50$, $t = 2.3$, and $u = \frac{N}{kr}$. For the specific set of parameters, $\left(-\frac{d}{k} - \left(\frac{1}{ur}\right)^{w-1} \cdot e^{-\frac{1}{u}}\right)$ is approximately 0. Thus the right part is approximately $e^{\frac{8k-2.3}{w+r-1}}$ which is a large value. Although this is only an asymptotic bound proven in Theorem 5, it shows an appropriate value always exists for a given specific set of parameters.

7.7.2 Shared key discovery

After distributed key selection, with high probability there exists a common key shared among all members of two neighboring communities C_i and C_{i+1} (on a multi-hop path) with $w = |C_i| + |C_{i+1}|$ nodes. However, these nodes have to figure out what that key is by a local key discovery (LKD) protocol.

In node-based key management, on a segment $C_i \leftrightarrow C_{i+1}$ of a multi-hop path, the node C_i needs to share a key with its next stop C_{i+1}. For community-based key management, C_i, C_{i+1} become partial trust communities, either of them is comprised of multiple nodes. The following LKD protocol can establish a common key shared between these two neighboring communities.

1. The RtREP node c_i (of the community C_i) broadcasts a Key Sharing Message (KSM) in the following format:

   $$\langle KSM, seq\#, i, i+1, \textit{key-ring-info} \rangle$$

   where $seq\#$ uniquely identifies each connection (e.g., the same source $seq\#$ used in RtREQ), i is the community index, and *key-ring-info* is to advertise key ring information [46]. That is, node c_i selects a random nonce α, then generates a ciphertext ring $\{E_{K_1}(\alpha), \ldots, E_{K_k}(\alpha)\}$ using every key in its key ring to encrypt the random nonce. *key-ring-info* is in the following format

   $$\langle \alpha, [I_{K_1}, E_{K_1}(\alpha)], \ldots, [I_{K_k}, E_{K_k}(\alpha)] \rangle$$

   In this work the size of α and E_{K_j} is 80 bits, and the size of I_{K_j} is 16 bits. Since no external adversary can produce a valid E_{K_j}, E_{K_j} is the cryptographic proof to show that the sender really has the corresponding key K_i in its key ring.

2. Any community member in C_i, C_{i+1} should find out those keys in its own key ring colliding with c_i's key ring. If there is key collision, then the key indexes I_{K_j} match, and the encrypted ciphertexts $E_{K_j}(\alpha)$ also match. An external adversary cannot forge $E_{K_j}(\alpha)$, thus the ciphertext is a proof that both key colliding parties really know the hidden K_j.

3. The community member then sends back a KSM_ACK message in a similar format

$$\langle KSM\_ACK, seq\#, i, i+1, \textit{key-ring-info} \rangle$$

where this time *key-ring-info* only holds information for those colliding keys. This *key-ring-info* field is much smaller:

$$\langle \beta, [I_{K_1}, E_{K_1}(\beta)], \ldots, [I_{K_{k'}}, E_{K_{k'}}(\beta)] \rangle$$

where k' is the number of colliding keys ($k' < k$ or $k' \ll k$). To resist replay attack, another nonce $\beta \neq \alpha$ is selected, thus a node not knowing K_j cannot produce a valid ciphertext proof $E_{K_j}(\beta)$.

4. Node c_i collects all KSM messages from C_i and C_{i+1}. It calculates the appearance frequency of each key, then chooses the key with highest frequency to be the key shared between community C_i and C_{i+1}. By DKS and Theorem 5 we know that a key shared by a community (of reasonable size) does exist with high probability. In the protocol we simply choose the key with highest frequency as an approximation of this key. Let's denote the key as $K_{i,i+1}$.

5. Finally node c_i broadcasts a KSM_KEY message in the following format:

$$\langle KSM\_KEY, seq\#, i, i+1, I_{K_{i,i+1}} \rangle.$$

Recall that this key distribution protocol only succeeds with high probability. It is possible that a community member fails to have the selected key $K_{i,i+1}$ in its key ring. This member sees this fact upon receiving the KSM_KEY message and resets its membership flag accordingly. This node is no longer a community member because it cannot decrypt any routing and data packet protected by the community key $K_{i,i+1}$.

7.7.3 Attacks against community-based key management

Since distributed key selection is done independently on each autonomous node, an attacker cannot interfere with other node's decisions. However, the shared key recovery phase is more vulnerable to adversary's attacks. In a community C_i, if the RtREP node c_i is an internal adversary, it will seek to decrease the community size by making the community keys shared by as few nodes as possible.

First, an adversarial RtREP node c_i may not send out the first KSM message to start the shared key discovery operations. But this will result in a collapse of the route because the protocol execution is incomplete and secure data forwarding is not feasible so far. As we described previously, such "all-or-nothing disruption" will finally incur another route discovery operation and relieve c_i from its control of the current route.

Second, c_i can try several methods, such as choosing keys with rare frequency among all KSM messages, to minimize the size of related communities. However, the KSM messages are overheard by all local nodes including all good nodes. Upon simple counting, they can detect obvious inconsistency between c_i's decision and the real situation. If the inconsistency is considered as anomaly, each good detector sends out a KSM_ANOMALY report

$$\langle KSM\_ANOMALY, seq\#, i, i+1, [I_{K_{i,i+1}}, freq], proof \rangle$$

where $I_{K_{i,i+1}}, freq$ are the accused key index and its observed frequency according to the anomaly reporter, and *proof* is the list of $[I_{K_j}, freq']$ pair such that I_{K_j} appears with a larger frequency $freq' > freq$ according to the reporter. If the upstream or downstream RtREP node does agree to the report, it will send RtERR notification to the source or the destination, thus incurs another route discovery operation and relieve c_i from its control of the current route. If the upstream or downstream RtREP node is a colluding internal adversary, the countermeasures are discussed in Section 7.9. In a nutshell, this anomaly report mechanism serves to accomplish the same "all-or-nothing disruption" effect against the compromised link. It deters internal adversaries from choosing obviously infrequent keys from KSM messages, and helps to maintain communities of reasonably large size.

7.8 Applying community-based communication in ad hoc routing protocols

In the presence of internal adversary, the data forwarding service of ad hoc routing protocols is vulnerable to disruption attacks. Here we present two exemplary protocols to demonstrate the usefulness of community-based communication. The first one is a general on demand routing protocol AODV. We do not choose DSR because it is relatively easy to implement node disjoint multi-path routing in source routing schemes [98]. Mobile nodes can simply check the source routing list to identify disjoint or conjoint paths. Thus route disruption caused by internal adversaries can be mitigated by disjoint multi-path routing, which is an alternative countermeasure other than our community-based communication. In AODV, it is relatively hard to realize node disjoint multi-path routing due to the nature of distance vector routing [91]. In

this case community-based communication plays a more important role in countering internal adversaries.

The second one is an on demand routing protocol ANODR with routing privacy support [78]. In order to provide anonymity and location privacy services, ANODR does not rely on intrusion detection, explicit authentication, and neighborhood discovery protocols. By selecting such a routing protocol, we seek to show that partial trust community is a general concept that is applicable to a diverse range of application domains.

7.8.1 Community-based AODV

As described previously, community-based communication does not change AODV's RtREQ and RtREP procedures except adding a 128-bit onion field to both RtREQ and RtREP packets for obtaining cryptographic onions to be used in periodical polling.

After source/destination onions used in polling are ready, the AODV source is responsible to keep the on demand route alive because it knows whether there is further data transmission. For every T_{poll}, the source sends out a separated POLL message if there is no data transmission, or piggybacks the short POLL message to the data packet otherwise. The source onion in POLL message will be decrypted at each stop and forwarded to the destination in the strict order embedded in the onion. If an adversarial forwarder drops the onion or does not peel off its onion layer, this onion is lost and then replying the polling message is impossible.

Upon receiving a POLL message, the AODV destination replies with a POLL_REP message holding the destination onion. Like the POLL message, this onion in POLL_REP will be decrypted at each stop and forwarded to the source in the strict order embedded in the onion. If the source fails to receive the POLL_REP within timeout, then it regards the route as broken and initiates route discovery to find new routes. Hence the

adversarial nodes are only given the chance to do “all-or-nothing disruption”, that is, they must either completely break the route, or do not interfere the polling process.

Partial trust community is maintained by monitoring POLL_REP messages. The first POLL_REP is sent right after RtREP, such that communities can be established. A node overhearing three consecutive POLL_REP messages should set its membership flag for the connection. The recorded community index is from the hop_count field of the second POLL_REP message among the three.

To establish a community key between every two neighboring communities en route, the key management protocol specified in Section 7.7 must be executed right after the first POLL_REP message. After keys are established between every neighboring communities, encrypted and authenticated data packets are forwarded from one community to the next en route.

7.8.2 Community-based ANODR

Similar to community-based AODV, community-based ANODR does not change ANODR’s RtREQ and RtREP procedures. In ANODR the destination already has the onion that can be used in periodic polling. To obtain the onion used by the source, a 128-bit field is added to RtREP packets. Like community-based AODV, the source knows whether it has further data to send. Thus the source is responsible to keep the on demand route alive. The POLL, POLL_REP, and data forwarding design is similar to community-based AODV.

ANODR has three variants. For fair comparison we will compare our work only with “anonymous-only ANODR”. This is because packet flows in community-based forwarding are traceable due to periodic polling. In current design it is inappropriate to implement dummy polling or to enforce neighborhood traffic mixing in community-based forwarding.

7.9 Discussions

Why not using Blundo's polynomial KPS scheme? Unlike the probabilistic KPS scheme used in this chapter, Blundo's polynomial KPS scheme [21] is a *deterministic* scheme for group key agreement. Any node group of limited size can meet together and agree on a key deterministically. It is known that Blom's scheme is the special case of Blundo's schem when group size is 2. We do not use Blundo's scheme because our adversary model includes active internal adversary. Any polynomial based scheme using > 1 degree polynomials, including Blundo's scheme, is not secure against internal adversaries who lie about their secret shares. Unlike CFF family, any two honest community nodes cannot agree on a key in Blundo's scheme if there is a single active internal adversary in the community. To find a liar, each share holder must present zero knowledge proof to prove the validity of its share. Since ZK proofs are public key based, it conflicts with the motivation of introducing KPS into the system design — why didn't we simply use key exchange schemes based on public key cryptography at the very beginning?

Therefore, we decide to rely on a more probabilistic approach — it is unlikely our probabilistic approach based on CFF family will either return the best result (i.e., all neighboring community nodes agree on a key) or the worst result (i.e., no key is agreed because of a single liar). We are also pursuing further study on mathematic properties of CFF family.

Overheads in community-based key management To establish a common key shared between two neighboring communities, several new message types, namely KSM, KSM_ACK, KSM_KEY, KSM_ANOMALY, are proposed to furnish the service. Among all kinds of such messages, the KSM message incurs significant communication overhead due to its long *key-ring-info* field. The other messages, namely

KSM_ACK, KSM_KEY, KSM_ANOMALY, are relatively short because they only hold information about a few colliding keys. Thus the associated communication overhead is also reasonably small.

The size of KSM's *key-ring-info* field is determined by k, the system parameter denoting the size of each node's key ring. Associated computational overhead is k symmetric key encryptions on a KSM sender site, but only a few on a KSM_ACK site for those colliding keys. By our calculation in Section 7.7.1, k is chosen to be 50 in this work. Recall that in Theorem 5 the increment in k exponentially increases T, namely the size of key sharing community. Thus k can be potentially increased to hundreds in order to build larger communities. With a larger k, a KSM message is of the size of hundreds/thousands of bytes. Since this message is transmitted once per link, the overall overhead incurred is equivalent to that of transmitting an end-to-end message of similar size between the source and the destination.

Trade-offs between routing integrity and routing privacy In order to ensure routing privacy, ANODR reveals neither hop count to the source or the destination, nor the per-hop keys to nodes other than RtREP nodes. However, ANODR is vulnerable to route disruption attacks. The community ANODR trades routing privacy for routing integrity. Consider each single hop on a multi-hop path, data forwarding service is compromised in node-based forwarding scheme if either end of the hop is compromised. In community-based communication, data forwarding service is backed up by all good members in the two neighboring communities. However, this routing integrity gain comes with the cost of routing privacy degradation. (1) If either end (i.e., RtREP node) of a single hop is compromised, then routing privacy on the hop is compromised in node-based forwarding schemes. This is also true in the community-based communication. (2) If neither end of a single hop is compromised, then routing privacy is protected in node-based forwarding schemes. However, in community-based

communication other community members are also given the community secret key. If one of them is compromised, then routing privacy on the hop is compromised. (3) In community-based ANODR, a community member is given the hop count between itself and the source or destination. This facilitates community-based key management (since w is minimized as a community key is only needed between communities with consecutive indexes), but compromises location privacy to some degree.

7.10 Evaluation of Community-based robust routing

We evaluate our community-based forwarding with both AODV and ANODR. With AODV protocol, we focus on internal adversary problem and demonstrate the performance gain by community-based forwarding scheme. For ANODR, we also show that community-forwarding mechanism further enhances the security of anonymous routing scheme.

7.10.1 Simulation Environment

We implement community-based forwarding scheme in QualNet [127], a detailed packet- level network simulator. For our simulations, we use CBR (Constant Bit Rate) application, UDP/IP (User Datagram Protocol/Internet Protocol), AODV (Ad-hoc On-Demand Distance Vector) routing protocol, IEEE 802.11 MAC and physical channel based on two-ray ground propagation model. For network device parameters, we use 2Mbits/sec channel capacity (i.e., bandwidth) and 250 meter power range. To simulate node mobility, we use random waypoint model [73].

The results are averaged over several simulation runs conducted with various random seeds. In each simulation scenario, 50 nodes are randomly placed within a 1500m$\times$300m field. For each CBR session, data packets of 512 bytes are generated

in a rate of 4 packets per second. The source-destination pairs are chosen randomly from all the nodes. During 15 minutes simulation time, average 10 short-lived pairs are maintained.

To evaluate community-based forwarding, we define a metric *forwarding percentage*. Forwarding percentage is calculated as the number of data packets being forwarded by a protocol under certain conditions normalized on an ideal situation. In our experiments, the ideal situation is obtained by running AODV without mobility and attacks. When the forwarding percentage is larger than 1, the protocol in question forwards more packets than AODV in the ideal case. When the forwarding percentage is smaller than 1, the protocol in question forwards less packets than AODV in the ideal case For example, 0.9 means that 10% of supposedly forwarded packets are not actually forwarded due to internal adversaries or other reasons. we also use following metrics to measure routing performance. *(i) packet delivery ratio*: the ratio between the number of data packets received and those originated by the sources. *(ii) routing overhead*: total number of routing control packets. Each hop-wise transmission of a routing packet is counted as one transmission. *(iii) average end-to-end packet latency*: the time from when the source generates the data packet to when the destination receives it. This includes: route acquisition latency, processing delays at various layers of each node, queueing at the interface queue, retransmission delays at the MAC, propagation and transfer delay.

We evaluate our community forwarding based AODV (denoted as AODV+CF w/ PROBE in the figures and mentioned as AODV+CF in this section for simplicity) and community forwarding based ANODR (denoted as ANODR+CF w/ PROBE in the figures and mentioned as ANODR+CF in this section for simplicity) with comparison to original AODV and ANODR. To clearly demonstrate the impact of end-to-end probing

community- maintenance mechanism, we also compare AODV+CF with AODV+CF without probing mechanism (denoted as AODV+CF w/o PROBE).

We implement KSM, KSM_ACK, KSM_KEY messages to simulate community key exchange. The size of "key-ring-info" field in a KSM message is set to 500 bytes which can hold fifty 80-bit keys. The same field in a KSM_ACK message is set to 50 bytes holding much less colliding keys. Any potential member sets an incomplete-key flag once 3 consecutive POLL_REPs are overheard for a connection. However, the key exchange success probability between two neighboring communities is only 0.9, hence a community has probability 0.81 (0.9^2) to share keys with both upstream and downstream communities. This condition is approximated by setting community membership flag with 0.81 probability on each potential member.

Our simulation will investigate (1) impact of internal adversaries on the performance and the resilience of community forwarding against rushing attack; and (2) impact of node mobility on community-forwarding scheme. In the first simulation study (Figure 7.7 to 7.11), we fix the node mobility with minimal speed = 2m/s, maximum speed = 20 m/s and pause time = 30s. We vary the percentage (p) of internal adversaries that perform black hole attacks from 0 to 20% (e.g., if p = 10, 5 nodes (0.1 * 50 nodes) are adversaries). In our second experiment (Figure 7.12 to 7.15), we fix the attacker ratio to 0.0 and 0.05 (2 or 3 nodes out of 50 nodes are internal adversaries) respectively, and vary the node mobility from static to maximum speed 20 in step of five while the minimum speed for mobile cases is fixed at 2 m/s. To measure the forwarding percentage, the baseline is the total number of forwarded packets of AODV in a static network without any attackers.

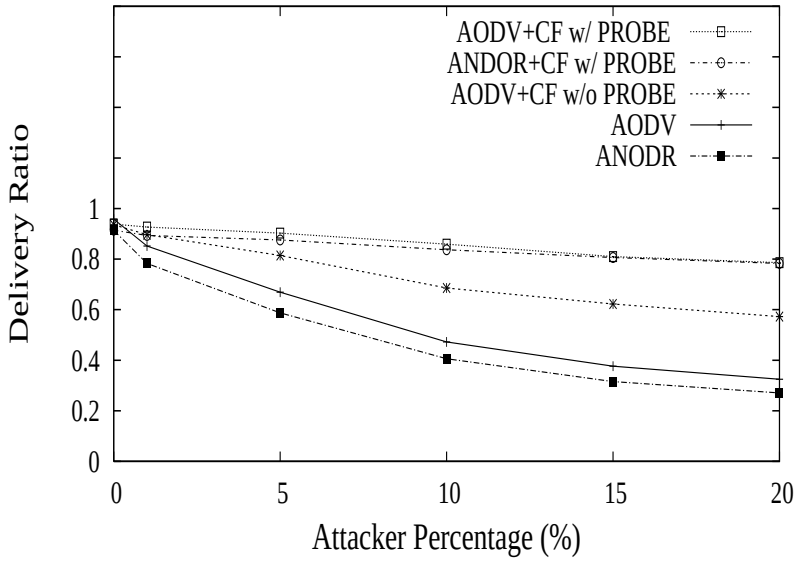

Figure 7.7: **Data Packet Delivery Ratio**

7.10.2 Simulation Results

Figure 7.7 illustrates the delivery ratio as function of increasing attacker ratio in the network. The figure shows three interesting results: First of all, the impact of internal adversary on AODV's performance is surprisingly significant. With 5% of adversaries (i.e., 2 or 3 nodes out of 50 nodes), the performance degrades more than 20%. ANODR also suffers from performance drop in the presence of attackers. Secondly, community-forwarding scheme notably improves the delivery ratio. The performance improvement of AODV+CF and ANODR+CF is up to 130% compared to AODV and ANODR. The incurred extra forwarding as demonstrated in Figure 7.8, however, is less than 22% even with higher attacker ratio. Lastly, community maintenance mechanism is essential to achieve high performance. Without probing, AODV+CF w/o PROBE suffers from stale partial-trust zone nodes which unnecessarily forward data packets that can not be received by downstream nodes.

Figure 7.8 illustrates the percentage of forwarded packets compared to that of AODV without adversaries. Notably, this result shows that the impact of internal

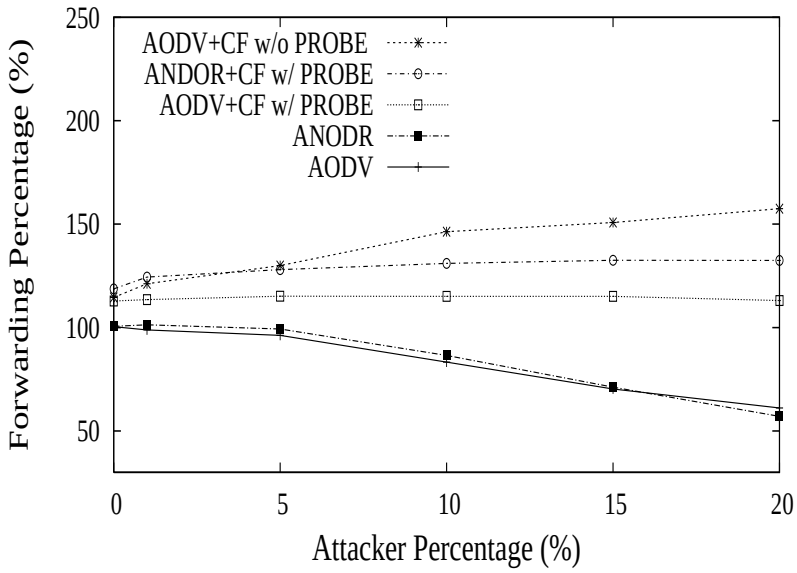

Figure 7.8: **Forwarding Percentage**

adversary to original AODV and ANODR is proportion to their presence, which is expected. For example, with 10% of adversary in the network, only 80% of packets is successfully forwarded and 20% of packets is dropped at those attackers. As an internal adversary node rushes to be an intermediate node, one adversary node possibly affects multiple paths. Due to multi-hop data transmission, the final delivery ratio is more seriously damaged by those adversaries as shown in 7.7. Moreover, we should note that extra forwarding overhead by community-based forwarding scheme keeps fairly stable and small (AODV+CF and ANODR+CF) in spite of increasing percentage of attackers. For AODV+CF and ANODR+CF, extra forwarding overhead exists even without adversary due to node mobility, hidden terminal and packet collisions. In the presence of packet collision and node mobility, a node in community may fail to overhead data forwarding from its core node even though the core already has successfully forwarded, and thus the node forwards the packet redundantly. Without probing maintenance, the AODV+CF w/o PROBE shows increasing forwarding overhead as number of attackers grows. High extra forwarding overhead is up to 60%. This is

because the stale community nodes while moving away from the community may still hear upstream transmitting, thus continuously transmit unnecessary data packets that can not be heard by down stream nodes and may already be transmitted by other community nodes. The more attackers, the more staled nodes forward packets. This impact of staled community member can also be observed in Figure 7.7.

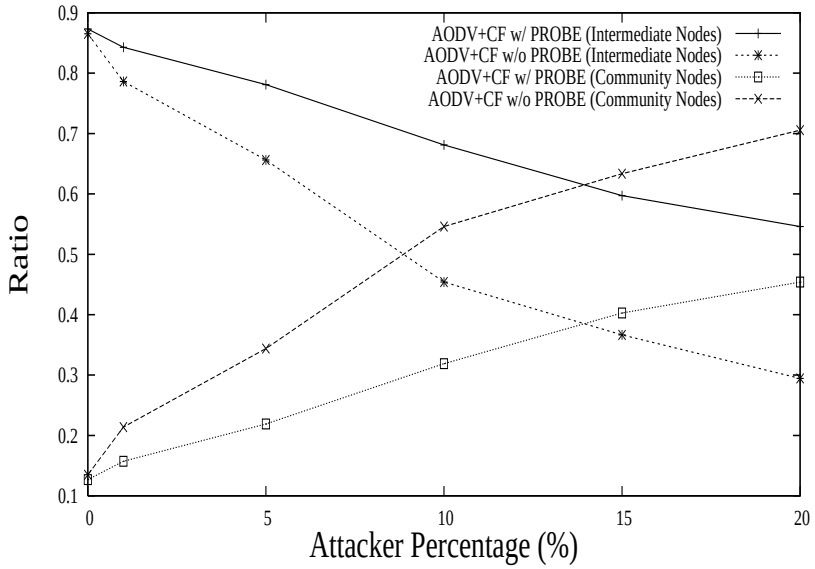

Figure 7.9: **Forwarding by Intermediate or Community Nodes**

Figure 7.9 shows the portion of intermediate nodes (INT) or community nodes (COM) that actually performs packet forwarding. The results show both AODV+CF and AODV+CF w/o PROBE schemes. It is clear that with increasing attacker ratio, the intermediate nodes fail more in forwarding, while the community nodes forward more packets to make amends for packet drops at attackers. For AODV+CF w/o PROBE scheme, the figure shows that more community nodes forward packets than nodes in AODV+CF. As explained previously, many forwarding is performed by stale community nodes. Transmissions by the stale nodes refrain qualified community node from forwarding thus reduce the packet delivery ratio (Figure 7.7), while generating unnecessarily or redundant packets which will never be received by the downstream nodes.

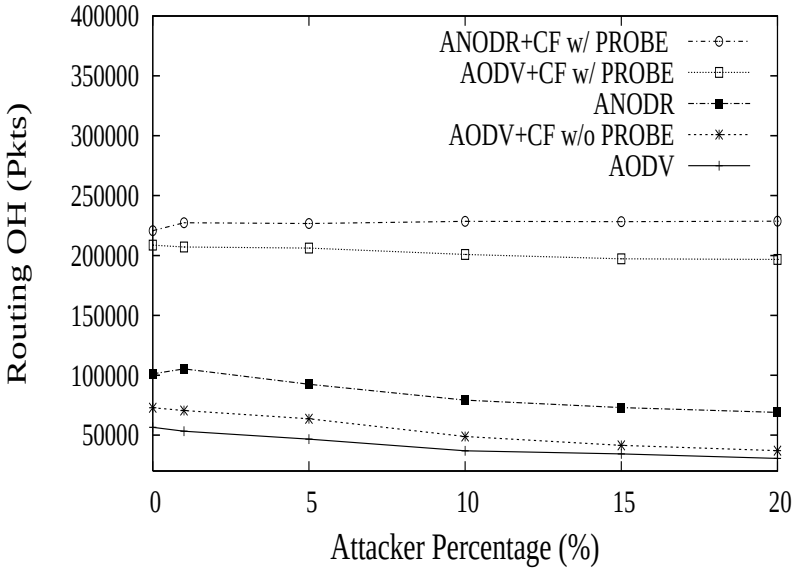

Figure 7.10: **Routing Overhead**

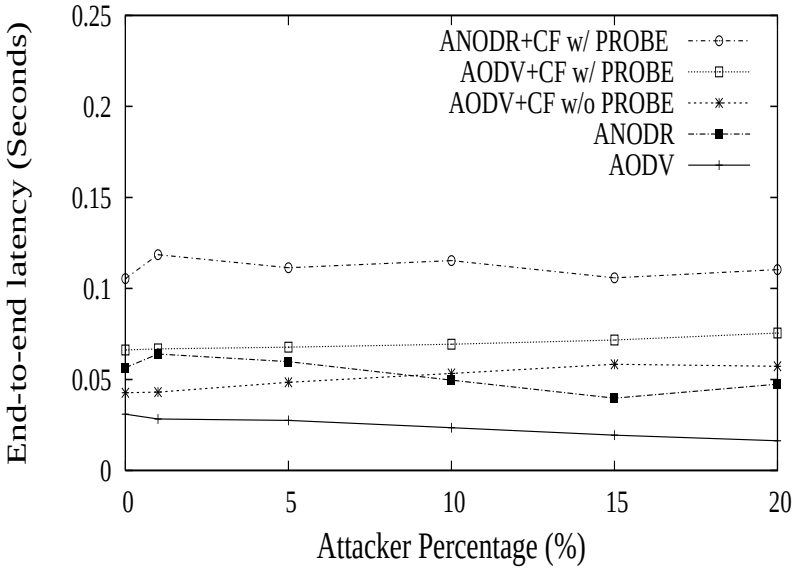

Figure 7.11: **End-to-end Latency**

Figure 7.10 shows the routing overhead. It clearly illustrates that AODV+CF and ANODR+CF trade high routing overhead to set up a shared key among community nodes for high delivery ratio and throughput. The total overhead paid for high packet delivery is about 200% of the standard AODV and ANODR. However, consistent to Figure 7.8, the routing overhead is not affected by the number of attackers. Without periodical probing, AODV+CF generates less overhead as expected (AODV+CF w/o PROBE).

Figure 7.11 shows the average end-to-end data packet latency with the increase of attacker ratio. The figure shows that community-based forwarding tends to generate high end-to-end latency due to key exchange overhead and extra community maintenance overhead. But the latency is not affected by the number of attackers. The figure also shows a decreasing trend in AODV and ANODR which is largely due to the fact that packets delivered far away have more chances to be dropped by an attacker when more and more attackers exist in the network.

Figure 7.12 to 7.15 show the impact of node mobility on community-forwarding schemes. The baseline for calculating the forwarding percentage is the total number of forwarded packets with AODV in static network without adversary nodes.

Figure 7.12 shows the delivery ratio when no attackers presents. It shows that community-based forwarding (both AODV+CF and ANODR+CF) slightly degrades the delivery ratio than the standard ones in this non-threatened environment. This is because community-based forwarding incurs extra overhead to establish community keys and to maintain the communities. However, the performance degradation is very small. ANODR+CF further reduces a little in data delivery due to the extra overhead and lack of optimization in order to achieve anonymity,

Figure 7.13 demonstrates that when there is mobility, AODV and ANODR forward less data packets comparing to AODV in the ideal condition (static network and

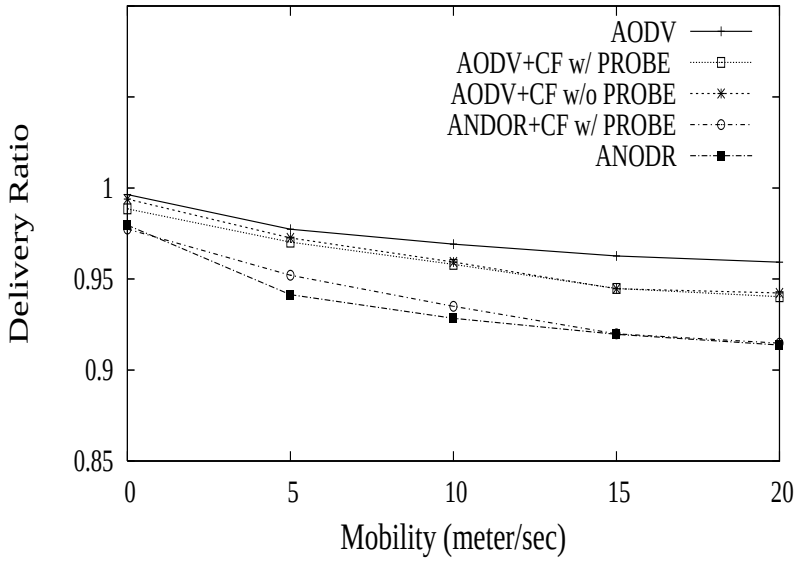

Figure 7.12: **Packet Delivery Ratio with Attacker Ratio = 0%**

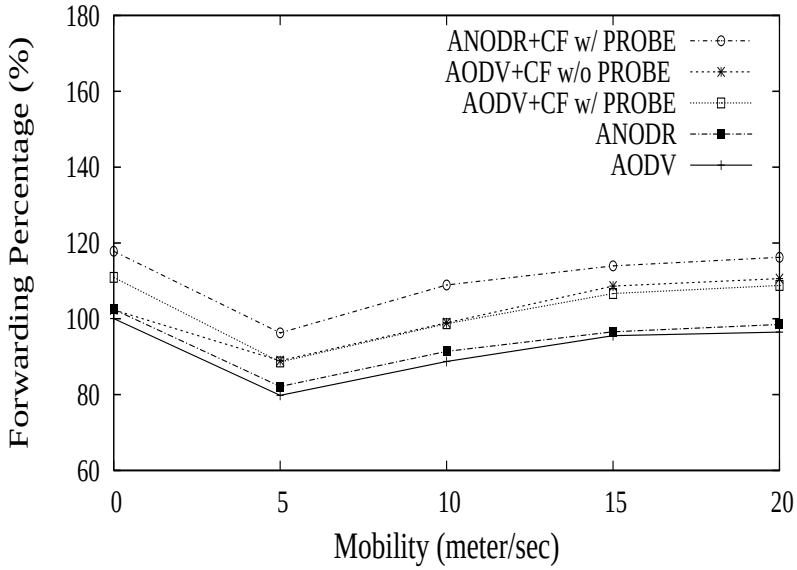

Figure 7.13: **Forwarding Percentage with attacker ratio = 0%**

no adversaries). AODV+CF and ANODR+CF forward more data packets than their counterparts. The extra overhead comes from redundant community forwarding, but the overhead does not exceed 20% and the ratio keeps stable with node mobility.

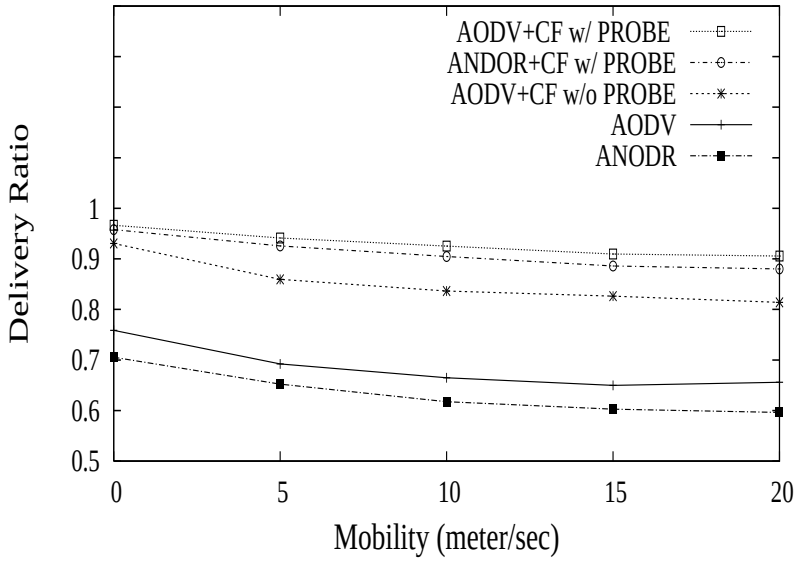

Figure 7.14: **Delivery Ratio with attacker ratio =5%**

Figure 7.14 and 7.15 show that community- forwarding schemes perform effectively with various node mobility in the presence of internal adversary nodes. With 5% adversary nodes, community-based forwarding schemes improve delivery ratio more than 10%. Figure 7.14 shows that ADOV and ANODR degrades to a delivery ratio lower than 80% while AODV+CF and ANODR+CF still keep at above 90%. And AODV+CF w/o PROBE also keeps at high delivered ratio even without community maintenance. Figure 7.15 shows that the ratio of extra forwarding overhead generated by AODV+CF and ANODR+CF remains at a level that is close to the one as depicted in Figure 7.13 where no adversary presents. Without probing, AODV+CF w/o PROBE unnecessarily forwarded more data packets due to the stale community members.

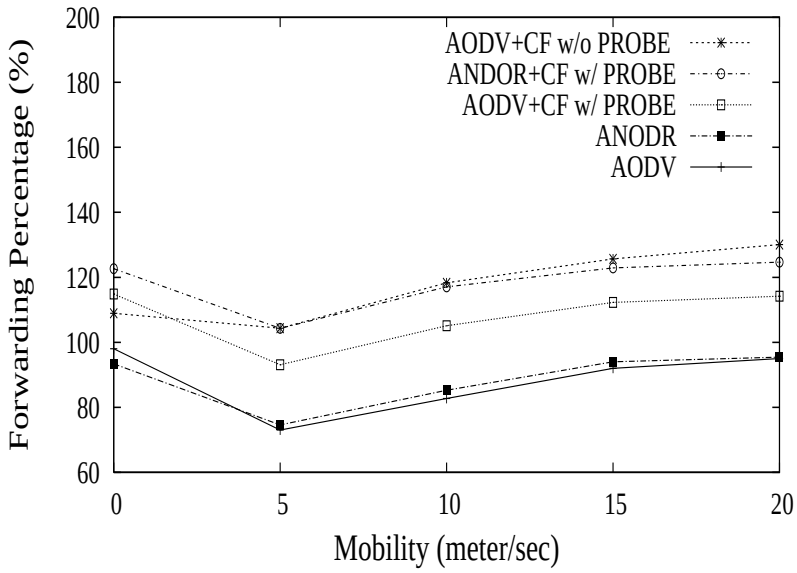

Figure 7.15: **Forwarding Percentage with attacker ratio =5%**

7.11 Summary

In this chapter we propose a new security mechanism, namely *community-based communication*, and evaluate how this new mechanism helps various ad hoc routing protocols to defend route disruption attacks. In particular, data forwarding service is used in this chapter as an example to demonstrate the usefulness of this concept. In contrast to conventional "per node forwarding", data forwarding service is provided by a sequence of local communities, each of them is comprised of multiple members that can provide the needed data forwarding service. The proposed scheme tolerates the presence of internal adversaries, and can be easily integrated with conventional routing scheme (e.g., AODV [108]) and anonymous routing scheme (e.g., ANODR [78]) with no major change required. We implement the new design in network simulators and study the new design's performance impact on underlying routing protocols.

As any new design always introduces new problems, we also address the new security challenges introduced by community-based communication. We study how to

defend attacks against community creation, configuration, and maintenance. We also devise a "community-based key management" scheme to provide cryptographic protections to community-based communication. In conventional node-based schemes, key management scheme is responsible to establish shared secret keys between any pair of nodes. Then such secret keys can be used in cryptographic protocols to ensure message privacy, message integrity, and other security services. We propose a *distributed* and *localized* key management scheme for community-based communication. The proposed scheme is friendly to routing performance because it is purely based on high-speed symmetric key cryptosystems.

CHAPTER 8

Related work

A dwarf standing on the shoulders of a giant
may see farther than the giant himself.

–Didacus Stella, circa AD60

8.1 Anonymity research

David Chaum initiated the research on anonymity and untraceability issues. A number of protocols to thwart content correlation and causality correction attacks, such as Web-MIXes [14], ISDN-MIXes [112], Stop-and-Go-MIXes [76], and many others, have been based on Chaum's anonymous E-mail solution: a network of *MIX*es [28]. They differ in mainly two aspects: network topology and traffic mixing strategy.

The topology of MIXes can either be organized as a random network (i.e., MIX-Net [113]) or a chain of MIX cascade [112]. MIX Cascade is not a scalable topology choice as number of network member increases. In TIMBA, MIX-Net is organized into hypercube, so that all network members become equivalent peers in terms of network topology. However, TIMBA/ANODR is *not* a porting of MIX-Net into mobile wireless networks. In MIX-Net, all senders, including adversarial senders, must know the entire MIX-Net topology. Such network topological knowledge is critical to providing anonymity protection in mobile networks. Revealing network topology to adversary is an anti-anonymity design rather than protecting mobile anonymity for the network.

An effort trying to directly port MIX-Net into mobile networks will establish a lot of single points of compromise in the network, hence not an apposite design choice.

Traffic mixing strategy includes introducing random latency, maintaining a threshold message pool with reordering and flushing, injecting dummy/decoy packets, and padding each packet to the same length or random lengths. Serjantov et al. [130] classify such mixing strategy into two major categories—simple MIXes and pool MIXes. Both categories use a threshold n and/or a time period t in traffic mixing. These two quantities are equivalent to the parameters d and T_{int} in TIBA, respectively. In this paper, traffic mixing can be considered as a practical actualization to achieve uniform distributions of transmission events. In TIBA or a one-hop TIMBA neighborhood, uniformly distributed broadcast communication and uniformly distributed transmission timer are ideal cases of traffic mixing. If identifiable transmission events are not uniformly distributed, degradation of anonymity design is inevitable [143]. Díaz et al. [36] proposed an information theoretic model for anonymity and analyzed how anonymity degrades in Onion Routing [118] and Crowds [119]. Serjantov and Danezis [129] presented a similar information theoretic model and analyzed pool mix as an example. The definition of perfect anonymity in this paper is consistent with related notions in these two information theoretic models. In the ideal case, sender and recipient anonymity in all three models are accomplished when senders and recipients are uniformly distributed over the entire set of network members.

Dining Cryptographer Network (DC-net) [29] is another proposal from Chaum. It requires pairwisely-shared 1-bit keys (or coin-flips) among network members. The parity of the coin-flips that a participant has seen is then announced to the public. Since each flip is announced twice, the total parity should be even. To send a message, a participant incorrectly states the parity seen. This causes the total parity to be odd, which indicates a message transmission. No one except the sender knows who sent the mes-

sage, unless all participants who flipped coins with the sender reveal their coin-flips among themselves. The routing in DC-net is not specified explicitly. Chaum's original paper suggested a closed ring. Hence a round-trip of a single message implements the network-wide broadcast needed in perfect anonymity (Figure 2.4). Dolev and Ostrovsky [43] devised a DC-net variant based on a network-wide multicast tree. The network-wide broadcast is then implemented by multicast in the network-wide tree. Both proposals implement perfect anonymity, but do not address timing analysis.

Many Internet applications also provide anonymity supports. Anonymous e-mailing is one of the earliest cases. Anonymous remailers in the Internet are loosely classified into four types. The Type 0 remailer (`http://anon.penet.fi/`) simply strips headers and resends the contents to the destination. This leaves a single point of compromise and failure in anonymity service. In 1996, legal pressure forced the operator to reveal actual sender-recipient correspondence, and the operator completely shut down the service. The Type 1 (or "cypherpunk") remailer [69] uses a set of remailers equipped with some MIX features. Like Chaumian MIX and α-TIMBA, the mail sender selects a chain of remailers, then forms an onion-like message using the public keys obtained by PGP. Each remailer could only see the address of the next remailer. Type 2 (or "Mixmaster") remailer [95] adds more MIX features to Type 1. An E-mail message is fragmented into a number of fixed-size packets. Each packet is delivered across the chain of remailers and finally re-assembled by the last remailer. Thus the system is less vulnerable to content correlation analysis. The extra cost is that new fragment/assemble software is needed in the remailer-net while anonymous E-mails can be delivered in a straight-forward manner in the previous Type 1 system. Type 3 (or "Mixminion") [34] uses a complex "swap" operation to defend tagging attack. It also requires a single-use "reply block" [57] that is used in anonymous email reply. However, all four types of remailers assume a fixed network as in MIX-Nets. The

sender has to schedule the entire routing complexity. This strategy is impractical in a mobile network.

Other Internet applications, such as World Wide Web, file transfer, and remote session, can be anonymized as well. The major difference between these Internet applications and Email is that these applications should handle bidirectional traffic interactively. In PipeNet [33], Onion Routing [118], and Freedom [146], ATM-like [7] virtual circuit is used in application overlay network to address bidirectional traffic for anonymous communications. Nevertheless, all these proposals still assume that the underlying MIX-Net is static and the source sender currently knows the entire MIX-Net topology. Besides, timing analysis is not fully considered in these designs.

There are some anonymous proposals focusing on concerns other than bidirectional traffic for interactive applications. Anonymizer [6] is an equivalence of Type 0 remailer for web services. Web-Mixes [14] employs strategies similar to Type 2 remailer. Tarzan's designers [53] studied how to instantiate Chaumian MIX-net as a network layer service on the Internet. Publius [141] applies threshold secret sharing scheme [131] to web access across many servers. This requires a new form of URL that can retrieve encrypted documents and can find the secret shares to reconstruct the decryption key. Reiter and Rubin [119] proposed Crowds system as an alternative to Chaumian MIX-net. Suppose we treat a multi-hop point-to-point path as a single logical link. The source sender randomly selects another node to be next logical forwarder, then sends the packet with encrypted real recipient's address attached to it. With *probability of forwarding* p_f the chosen forwarder randomly selects another node to be next logical forwarder, else it sends the packet to the real recipient directly. The expected number of intermediate forwarding is $1/p_f$ due to the nature of geometric distribution.

However, due to lack of uniformly distributed traffic pattern Crowds is vulnerable to content correlation and causality correlation analysis [10][143][36].

Anonymous document storage, a recently proposed anonymous application, is a slightly different problem from the mobile anonymity problem studied in this paper. In mobile anonymity, a sender peer wants to deliver some messages to its recipient peer via anonymous communications. The goal of such anonymous communications is to let them both see the messages. In contrast, anonymous document storage follows a client-server paradigm (i.e., they are not peers). A common feature of related proposals is to realize redundancy across a large set of nodes. In Eternity Service [5], a client disseminates his documents to certain number of servers which may only maintain the documents with certain probability due to censorship and other reasons. To retrieve the documents, the client broadcasts queries to the servers, and document delivery is achieved through unidirectional anonymous remailers. If large-scale data dissemination mechanisms (e.g., Usenet infrastructure [9]) are available, Eternity Service readily provides uncensorable services. In Freenet [32], a limited flooding protocol is used to find storage sites. Besides, the document owner encrypts the contents so that the storage sites do not know the contents. Free Haven [40] applies threshold secret sharing in document storage. A document is dispersed to distributed servers and any threshold shares of the dispersal can re-construct the original document. Hence the scheme tolerates a threshold number of adversarial or crashed servers. Besides, Free Haven employs a reputation system to control reliability issues in the distributed server set.

8.2 Security in mobile ad hoc networks

Recently many solutions are proposed for ad hoc routing schemes to resist active and passive routing attacks. We categorize these solutions into two general types: external attack countermeasures and internal attack countermeasures.

Nearly all proposed countermeasures can defend one or more forms of external attacks. To resist active route disruption, either public key based digital signatures or symmetric key based TESLA protocol [109][110] can be used to differentiate legitimate members from external adversaries. A major difference between the two approaches is of performance concerns. Hu et al. [65] illustrated a feasible resource consumption attack against digital signature based countermeasures. Because nodes in an ad hoc network may not have sufficient resources to verify a digital signature generated in a public key cryptosystem, an attacker can trivially pay little computational cost to flood a victim with packets containing random fake signatures, while signature verification can be prohibitively expensive for the victim. On the other hand, the advantage of digital signature is its simple key management assumption. As unforgeable cryptographic ID (e.g., certificate) can be assigned to every network member by an off-line authority, nodes' public keys can be securely exchanged by showing cryptographic IDs. This approach is used in ARAN [125] and ubiquitous network admission control [81]. In a symmetric key cryptosystem, key distribution is an essential problem that must be solved. Fortunately, recently there are many probabilistic key pre-distribution scheme (KPS) proposed for symmetric key cryptosystems used in ad hoc networks. In sensor networks, KPS is proposed in a series of recent publications [46][26][44][89] to replace public key exchanges. In mobile ad hoc networks, this subject is studied by Zhu et al. [150]. However, these proposals are designed only for key establishment between two single nodes. Community-based communication also explores probabilistic KPS in its key management module. It is designed for

community-versus-community communication. This differentiates our scheme from these existing proposals.

Defending internal attacks is very different from defending external attacks. Besides cryptographic means, an internal attack countermeasure must also rely on network-based mechanisms. In Packet Leashes, Hu et al. [66] explored geographical distance and timing difference to limit a node's data delivery capability in terms of temporal and spatial aspects. This mitigates route disruptions caused by "wormhole" attacks. Papadimitratos and Haas [103] studied a multi-path approach to mitigate route disruption attacks. By encoding data packets into erasure codes, the destination is able to recover the source's data upon receiving a threshold subset of encoding symbols that have been delivered along the multiple paths. Yang and Lu [144] proposed a secure AODV protocol based on "watchdog" mechanism [92] and probabilistic intrusion detection. In rushing attack studies, Hu et al. [67] proposed to form explicit local communities by a secure neighborhood discovery protocol. In a local community, RtREQ forwarding is randomized so that an RtREQ rushing attacker cannot dominate other members during the RtREQ phase. Route disruption is mitigated because the chance of selecting a rush attacker on a path equals the chance of selecting a good member. Awerbuch et al. [8] proposed a path evaluation and polling scheme to detect internal adversaries on an ad hoc route. The onion based polling scheme was used in ad hoc network security for the first time in this paper, but they assume that the keys shared between the source and the mobile forwarders are already established. This is not a realistic assumption in a mobile network. In contrast, we explore existing on-demand route discovery procedures and form the onions without such unrealistic key management assumptions. Moreover, the most significant difference between our scheme and Awerbuch's scheme is the latter one is designed for conventional node-versus-node communication.

Internal attacks and external attacks are correlated with each other. An internal ad-

versary may distribute an exposed cryptographic secret to external adversaries. Therefore, leaving single point of compromise and failure in a solution may lead to the final collapse of the entire security service. Unfortunately, as ad hoc networks rely on distributed cooperative algorithms to provide needed network services, it is a non-trivial challenge to devise fully distributed security solutions without single point of compromise and failure. In some secure routing proposals, a system cryptographic secret is replicated globally or among a node set of a considerable size. This leaves single point of compromise in the network. Basagni et al. [11] argue to protect the system secret using tamper resistance. In distributed trust, a system secret is protected by threshold cryptography [149] [81] [148] [126] [97]. Such a secret is partitioned into n shares, and any k of the n shares can be combined to provide the same cryptographic service as before. The overall service tolerates $(n-k)$ node failures and $(k-1)$ node compromises. In contact-based trust [68][139][138], ad hoc nodes do not trust another node unless there is other relatively sound proof. Such proof is normaly realized by physical contact or other secure means. This approach relies more on physical and social means to secure ad hoc networks.

CHAPTER 9

Summary and future work

It's never over till it's over.

–American proverb

In this dissertation we have proposed new anonymity models and anonymous routing schemes. For mobile wireless networks, we define the concept "mobile anonymity" as our design goal. We show that current anonymous communication schemes are not applicable to mobile wireless networks, and current routing schemes fail to provide anonymity protection. Our study is based on a comprehensive adversary model, which includes both external and internal adversary, as well as both passive and active adversary.

We propose an ideal model, *Time Interval and Multi-hop Broadcast based Anonymity (TIMBA)*, and prove that TIMBA ensures perfect mobile anonymity against passive external adversary. Then we devise ANODR as a practical balance between the ideal TIMBA model and the real world. ANODR is comprised of three variants, each of them trades off security guarantees with routing performance at different level. There is a significant distinction between TIMBA/ANODR and other schemes, such as anonymous schemes like MIX-Net and any existing routing scheme: TIMBA/ANODR is a unique scheme where *no node identity is used in data forwarding and routing*. This ensures perfect identity anonymity in mobile networks, even against internal adversary and active adversary.

We also study how to defend internal adversary and how to integrate ANODR with countermeasures against active adversary. We show that internal adversary can degrade mobile anonymity protection, and related countermeasures can be prohibitively expensive. To defend active adversary, we define the concept of "*partial trust community*" and realize a new community-based communication paradigm. We devise new protocols to create, configure, maintain partial-trust communities, as well as a pairwise community key agreement scheme to secure per-hop data forwarding. Since any newly added function always introduces new vulnerabilities, we also study how to defend new security attacks against our proposed schemes. We then design and simulate community-based AODV and community-based ANODR to verify the usefulness of the concept of partial trust community. These two exemplary secure routing protocols justify the observation that partial trust is a general design that can be explored in diverse application contexts.

There are many challenging subjects left to be addressed in the future. First, scalability questions are raised against ANODR and other on-demand routing protocols. Compared to existing on-demand routing protocols like DSR and AODV, it is relatively hard to add routing optimization supports to ANODR due to anonymity requirements. The challenge of designing an anonymous, scalable and practical routing protocol is not answered in this dissertation. In the near future, we should pursue scalable routing schemes that can protect network's mobile anonymity. If perfect mobile anonymity is indeed not achievable in a scalable routing protocol, we also need to identify the tradeoffs between security and scalability. Second, how to quantify mobile anonymity degradation caused by internal adversary is still an open challenge. We know that perfect protection against internal adversary is not possible. Like modern cryptography, we should speak of the infeasibility of breaking a security scheme rather than the notion of the impossibility of breaking the same scheme. So far it is not very clear how we can model damage caused by internal adversary in a quantifiable frame-

work, so that we can apply scientific or numeric approaches to show that the damage is actually below certain acceptable level (e.g., sub-polynomial). Third, anonymity protection is merely one aspect of all possible security protections needed in a network system. How to integrate an anonymity solution with other solutions is another open challenge. Previous security research has shown that composition of multiple security protections may not be secure (e.g., authentication and encryption [13][82]). In Chapter 7 we show there are indeed tradeoffs between mobile anonymity and routing integrity. Finally, performance is a critical evaluation criteria of any network security scheme, not all polynomial-time algorithms are useful in the real world even if they are considered efficient in algorithm design. We need to explore every chance to minimize computational and communication overhead incurred in a secure network scheme.

APPENDIX A

Underlying cryptography

A.1 Probabilistic computation model

Any practical security solution can only defend a corresponding adversary bounded by reasonable capability. There are perfectly secure systems (e.g., Shannon's perfect cipher [132]), but such solutions have been empirically proven as impractical in the real world. Therefore, modern security research has to find solutions to defend a relaxed but practical adversary. In modern cryptography, we speak of the *infeasibility* of breaking a cryptosystem rather than the historical notion of the *impossibility* of breaking the same system. A cryptosystem is good if the probability of breaking it is sub-polynomial.

Definition 6 ***(Negligible):*** *A function* $\mu : \mathbb{N} \rightarrow \mathbb{R}$ *is* negligible *if for every positive polynomial* $P(\cdot)$ *and all sufficiently large* n*'s,*

$$\mu(n) < \frac{1}{P(n)}.$$

□

The concept of one-way function is defined on polynomial relation between the input length and output length. Here we follow the common definition [54]:

Definition 7 ***(Bounded-Probability Polynomial Time, $\mathcal{BPP}$):*** *Let $M(x)$ be the random variable denoting the output of a probabilistic machine M. Let*

$$\mathsf{Pr}[M(x) = y] = \frac{|\{d \in \mathbb{Z}_2^{t_M(x)} : \ M_d(x) = y\}|}{r^{t_M(x)}}$$

where d is a truly random coin-flip, $t_M(x)$ is the number of coin-flips made by M on input x, and $M_d(x)$ denotes the output of M on input x when d is the outcome of its coin-flips.

We say that L is recognized by the probabilistic polynomial-time Turing Machine M if

- *for every $x \in L$ it holds that $\mathsf{Pr}[M \ accepts \ x] \geq \frac{1}{2} + \frac{1}{P(n)}$ for every polynomial $P(\cdot)$.*
- *for every $x \notin L$ it holds that $\mathsf{Pr}[M \ accepts \ x] \leq \frac{1}{2} - \frac{1}{P(n)}$ for every polynomial $P(\cdot)$.*

$\mathcal{BPP}$ is the class of languages that can be recognized by a probabilistic polynomial time Turing Machine. □

Definition 8 ***(Non-uniform polynomial time, $\mathcal{P}/\mathcal{P}oly$):*** *A* non-uniform polynomial time machine *is a pair $(M, \bar{a})$, where M is a two-input polynomial-time Turing machine and $\bar{a} = a_1, a_2, \ldots$ is an infinite sequence of strings such that their length $|a_n|$ is a polynomial of n. For every x, we consider the computation of machine on the input pair $(x, a_{|x|})$.*

The complexity class non-uniform polynomial time *(denoted $\mathcal{P}/\mathcal{P}oly$) is the class of languages L that can be recognized by a non-uniform sequence of polynomial*

time machines. Namely, $L \in \mathcal{P}/\mathcal{P}oly$ if there exists an infinite sequence of machines $M_1, M_2, \ldots$ satisfying the following:

- *There exists a polynomial $p(\cdot)$ such that for every n, the description of machine M_n has length bounded above by $p(n)$.*
- *There exists a polynomial $q(\cdot)$ such that for every n, the running time of machine M_n on each input of length n is bounded above by $q(n)$.*
- *M_n is defined as $M_n(x) \triangleq M(x, a_{|x|})$, that is, the Turing Machine that computes the input pair $(x, a_{|x|})$. For every n and every $x \in \{0,1\}^n$, machine M_n will accept x if and only if $x \in L$.*

□

It is proven by Adelman [2] that $\mathcal{BPP}$ is a subclass of $\mathcal{P}/\mathcal{P}oly$.

Theorem 6

$$\mathcal{BPP} \subseteq \mathcal{P}/\mathcal{P}oly$$

Therefore, if it is proven a cryptosystem is secure against $\mathcal{P}/\mathcal{P}oly$ adversary, then the same system is secure against $\mathcal{BPP}$ adversary.

The concept of one-way function is defined on polynomial relation between the input length and output length. Intuitively, it is realized by some special NP problems that are hard in all (or nearly all) cases, such that adversaries must exhaustively search all possible cases in a non-deterministic Turing Machine. As $P \stackrel{?}{=} NP$ is an open problem, the existence of one-way function is not proven. Yet a number of conjectured collections of one-way functions are routinely used in commerce and industry, such as Discrete Logarithm [38], RSA function [121], Rabin function [116], and Feistel Structure [50]. If one-way function does exist, it can generate *cryptographically strong*

pseudorandom ensembles, which are indistinguishable from *truly random ensembles* by any Turing Machine in polynomial time.

Here we follow the common definitions used in cryptologic research [54]:

Definition 9 ***(One-way Function):*** *Let $\mathbb{Z}_2^*$ denote binary strings with arbitrary positive length. A function $f : \mathbb{Z}_2^* \mapsto \mathbb{Z}_2^*$ is a (strong)* one-way *function if the following two conditions hold:*

1. Easy to compute*: There exists a deterministic polynomial-time algorithm A such that on input x it outputs $f(x)$, i.e., $A(x) = f(x)$.*

2. Hard to invert*: The probability to invert the function is negligible. That is, for every probabilistic polynomial-time algorithm A', every positive polynomial $P(\cdot)$, and all sufficiently large n,*

$$Pr[A'(f(U_n), 1^n) \in f^{-1}(f(U_n))] < \frac{1}{P(n)}$$

where U_n denotes a random variable uniformly distributed over $\mathbb{Z}_2^n$, and the auxiliary input 1^n gives the length of the desired output n in unary notation. In particular, $\in$ can be replaced by $=$ if f is bijective, and the auxiliary input 1^n is redundant if the one-way function is endomorphic $f : \mathbb{Z}_2^n \mapsto \mathbb{Z}_2^n$. □

One-way function can be used to generate cryptographically strong pseudorandom ensembles with the help of *hard-core predicate*. Goldreich and Levin [55] proved that b is a hard-core predicate of any one-way function if b is defined as inner product mod 2. Blum and Micali [20] proved that the following generator is cryptographically strong pseudorandom generator (CSPRG).

Definition 10 ***(Blum-Micali pseudorandom generator):*** *Let f be a family of trapdoor one-way permutations $f : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \mapsto \mathbb{Z}_2^n$. Let $b : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \mapsto \mathbb{Z}_2$ be a polynomial-*

time-computable hard-core predicate of f, and let $P(\cdot)$ be an arbitrary polynomial satisfying $P(n) > n$. Given a truly random inputs $s, x, y \in_U \mathbb{Z}_2^n$, the pseudorandom generator G is defined as $G(s) = \sigma_1\sigma_1 \cdots \sigma_{P(n)}$, where $s_0 \stackrel{def}{=} s$, and for every $1 \leq i \leq P(n)$ it holds that $\sigma_i = b(x, s_{i-1})$ and $s_i = f(y, s_{i-1})$. That is, the algorithm G proceeds as follows:

1. *Uniformly choose the seed $s_0 \in_U \mathbb{Z}_2^n$.*
2. *For $i = 1$ to $P(n)$ do $\sigma_i \leftarrow b(x, s_{i-1})$ and $s_i \leftarrow f(y, s_{i-1})$, where $x, y \in_U \mathbb{Z}_2^n$.*
3. *Output $\sigma_1\sigma_1 \cdots \sigma_{P(n)}$.*

G is a cryptographically strong pseudorandom generator. □

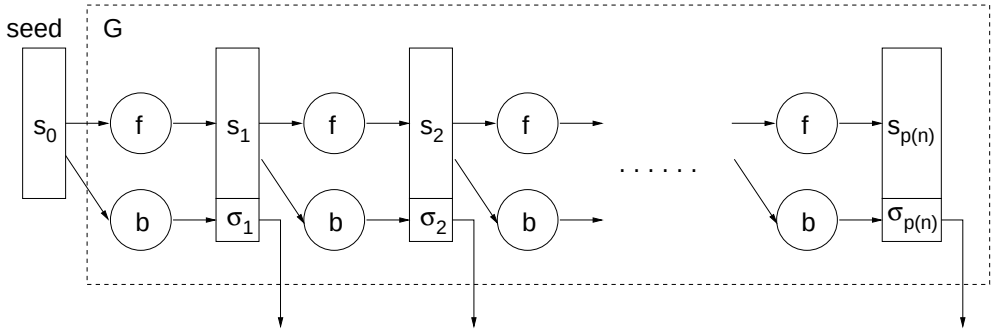

Figure A.1: **Blum-Micali pseudorandom generator** (A binary cryptographically strong pseudorandom generator)

The essential structure of Blum-Micali pseudorandom generator is depicted in Fig. A.1. If the one-way function is public, then we must output the depicted result in a reversed order $(\sigma_{P(n)}\sigma_{P(n-1)} \cdots \sigma_1)$. In this paper we assume that a sender and a recipient can generate arbitrarily long pseudorandom ensembles using a shared seed s_0 and Blum-Micali pseudorandom generator. Intuitively, given the fact f is one-way, we can conclude that no Turing-complete algorithm on earth can effectively see the difference between a truly random bit string and a bit string generated by CSPRG.

There are other practical CSPRGs that can be used by us:

- Blum-Blum-Shub generator [17] assumes factoring is a hard one-way function. Two principals share two secret large prime numbers p and q such that

$$p{\equiv}q{\equiv}3 \pmod 4.$$

  The product $n = p{\cdot}q$ can be made public. Blum-Blum-Shub generator is a pseudorandom bit generator that generates result B_i bit-by-bit. Here n_i is the block of bits between $B_{i{\cdot}128}$ and $B_{i{\cdot}128+127}$ as we currently use 128-bit route pseudonyms.

Initialization:

$X_0 = (seed)^2 \bmod n$

Loop:

For $i = 0$ to ∞

$X_i = (X_{i-1})^2 \bmod n$

$B_i = X_i \bmod 2$

- The pseudorandom number generator specified in X9.17 [4] assumes triple-DES is a hard one-way function. This generator generates result R_i block-by-block. Here n_i is comprised of two TripleDES blocks $R_{2{\cdot}i}$ and $R_{2{\cdot}i+1}$ because TripleDES uses 64-bit blocks and each of route pseudonym is currently 128-bit.

$$R_i = TripleDES[X_i{\oplus}TripleDES[DT_i]]$$

$$X_{i+1} = TripleDES[R_i{\oplus}TripleDES[DT_i]]$$

  where X_0 is the secret $seed$, and DT_i is a random value which could be the

system clock for single-party usage, or a synchronized random value for two-party usage.

A.2 Blom's Key Pre-distribution Scheme (KPS)

Blom proposed a key pre-distribution method that allows any pair of nodes in a network to be able to find a pairwise secret key [16]. As long as no more than λ nodes are compromised, the network is perfectly secure (this is called the λ-secure property).

Before network nodes join the network, an offline authority first constructs a $(\lambda + 1) \times N$ matrix G over a finite field $GF(q)$, where N is the size of the network. G is well-known public information. Then the offline authority creates a random $(\lambda + 1) \times (\lambda + 1)$ symmetric matrix D over $GF(q)$, and computes an $N \times (\lambda + 1)$ matrix $A = (D \cdot G)^T$ where $(D \cdot G)^T$ is the transpose of $D \cdot G$. D must be kept secret, or the cryptosystem is broken if D is revealed to adversary.

Because D is symmetric, it is easy to verify:

$$A \cdot G = (D \cdot G)^T \cdot G = G^T \cdot D^T \cdot G = G^T \cdot D \cdot G = (A \cdot G)^T.$$

This means that $A \cdot G$ is a symmetric matrix. If we let $K = A \cdot G$, we know that $K_{ij} = K_{ji}$, where K_{ij} is the element in K located in the i-th row and j-th column. We use K_{ij} (or K_{ji}) as the pairwise key between node i and node j. To carry out the above computation, nodes i and j should be able to compute K_{ij} and K_{ji}, respectively. This can be easily achieved using the following key pre-distribution scheme, for $k = 1, \ldots, N$:

1. store the k-th row of matrix A at node k, and
2. store the k-th column of matrix G at node k.

Therefore, when nodes i and j need to find the pairwise key between them, they first exchange their columns of G, and then they can compute K_{ij} and K_{ji}, respectively, using their private rows of A. Because G is public information, its columns can be transmitted in plaintext. It has been proved by Blom [16] that the above scheme is λ-secure if any $\lambda + 1$ columns of G are linearly independent. This λ-secure property guarantees that no nodes other than i and j can compute K_{ij} or K_{ji} if no more than λ nodes are compromised.

Note that any $\lambda + 1$ columns of G must be linearly independent in order to achieve the λ-secure property. Since each pairwise key is represented by an element in the finite field $GF(q)$, if the length of pairwise keys is 64 bits, then we should choose q as the smallest prime number that is larger than 2^{64}. Let s be a primitive element in $GF(q)$ and $N < q$. That is, each nonzero element in $GF(q)$ can be represented by some power of s. A feasible G can be designed as follows [90]:

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^N \\ s^2 & (s^2)^2 & (s^2)^3 & \cdots & (s^2)^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s^\lambda & (s^\lambda)^2 & (s^\lambda)^3 & \cdots & (s^\lambda)^N \end{bmatrix}$$

It is well-known that $s^i \neq s^j$ if $i = j$ (this is a property of $GF(q)$). Since G is a Vandermonde matrix, it can be shown that any $\lambda + 1$ columns of G are linearly independent when $s, s^2, s^3, \ldots, s^N$ are all distinct [90]. In practice, G can be generated by the primitive element s of $GF(q)$. Therefore, when we store the k-th column of G at node k, we only need to store the seed s^k at this node, and any node can regenerate the column given the seed.

REFERENCES

[1] M. Abe. Universally Verifiable MIX With Verification Work Independent of The Number of MIX Servers. In *EUROCRYPT'98, Lecture Notes in Computer Science 1403*, pages 437–447, 1998.

[2] L. Adelman. Two Theorems on Random Polynomial Time. In *Symposium on Foundations of Computer Science (FOCS)*, pages 75–83, 1978.

[3] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol. `http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-08.txt`, March 2003.

[4] American National Standards Institute. American National Standard X9.17: Financial Institution Key Management (Wholesale), 1985.

[5] R. J. Anderson. The Eternity Service. In *1st International Conference on the Theory and Applications of Cryptology (PRAGOCRYPT)*, pages 242–252, 1996.

[6] Anonymizer.com. Online Privacy Services. `http://www.anonymizer.com`.

[7] ATM Forum. Asynchronous Transfer Mode. `http://www.atmforum.org/`.

[8] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. In *First ACM Workshop on Wireless Security (WiSe)*, pages 21–30, 2002.

[9] A. Back. The eternity service. *Phrack Magazine*, 7(51), September 1997.

[10] A. Back, U. Möller, and A. Stiglic. Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems. In I. S. Moskowitz, editor, *Fourth Internationl Workshop on Information Hiding (IH'01), Lecture Notes in Computer Science, 2137*, pages 245–257, 2001.

[11] S. Basagni, K. Herrin, E. Rosti, and D. Bruschi. Secure Pebblenets. In *MobiHoc*, pages 156–163, 2001.

[12] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In H. Krawczyk, editor, *CRYPTO'98, Lecture Notes in Computer Science 1462*, pages 26–45, 1998.

[13] M. Bellare and C. Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In T. Okamoto, editor, *ASIACRYPT'00, Lecture Notes in Computer Science 1976*, pages 531–545, 2000.

[14] O. Berthold, H. Federrath, and S. Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In H. Federrath, editor, *DIAU'00, Lecture Notes in Computer Science 2009*, pages 115–129, 2000.

[15] O. Berthold, A. Pfitzmann, and R. Standtke. The disadvantages of free MIX routes and how to overcome them. In H. Federrath, editor, *DIAU'00, Lecture Notes in Computer Science 2009*, pages 30–45, 2000.

[16] R. Blom. An Optimal Class of Symmetric Key Generation System. In T. Beth, N. Cot, and I. Ingemarsson, editors, *EUROCRYPT'84, Lecture Notes in Computer Science 209*, pages 335–338, 1985.

[17] L. Blum, M. Blum, and M. Shub. A Simple Unpredictable Pseudo-Random Number Generator. *SIAM Journal on Computing*, 15(2):364–383, 1986.

[18] M. Blum, P. Feldman, and S. Micali. Non-Interactive Zero-Knowledge Proof Systems and Applications. In *20th Symposium on the Theory of Computation (STOC)*, pages 103–112, 1988.

[19] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. In *Symposium on Foundations of Computer Science (FOCS)*, pages 112–117, 1982.

[20] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. *Society for Industrial and Applied Mathematics (SIAM) Journal on Computing*, 13(4):850–864, 1984.

[21] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-Secure Key Distribution for Dynamic Conferences. In E. F. Brickell, editor, *CRYPTO'92, Lecture Notes in Computer Science 740*, pages 471–486, 1993.

[22] D. Boneh and P. Golle. Almost Entirely Correct Mixing With Application to Voting. In V. Atluri, editor, *9th ACM Conference on Computer and Communications Security (CCS'02)*, pages 68–77, 2002.

[23] M. Brown, D. Cheung, D. Hankerson, J. L. Hernandez, M. Kurkup, and A. Menezes. PGP in Constrained Wireless Devices. In *USENIX Security Symposium (Security '00)*, 2000.

[24] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast Security: A Taxonomy and Some Efficient Constructions. In *IEEE INFOCOM*, pages 708–716, 1999.

[25] A. C.-F. Chan. Probabilistic Distributed Key Pre-distribution for Mobile Ad hoc Networks. In *IEEE International Conference on Communications (ICC)*, 2004. Wireless Networking Symposium: WN04-4.

[26] H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In *IEEE Symposium on Security and Privacy*, pages 197–215, 2003.

[27] D. Chaum and T. Pedersen. Wallet Database with Observers. In *CRYPTO*, pages 89–105, 1993.

[28] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[29] D. L. Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[30] D. L. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In S. Goldwasser, editor, *CRYPTO'88, Lecture Notes in Computer Science 403*, pages 319–327, 1989.

[31] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In *Symposium on Foundations of Computer Science (FOCS)*, pages 335–344, 1985.

[32] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In H. Federrath, editor, *DIAU'00, Lecture Notes in Computer Science 2009*, pages 46–66, 2000.

[33] W. Dai. PipeNet 1.1. `http://www.eskimo.com/~weidai/pipenet.txt`, 1996.

[34] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *IEEE Symposium on Security and Privacy*, 2003.

[35] Y. Desmedt and K. Kurosawa. How to Break a Practical MIX and Design a New One. In *EUROCRYPT'00, Lecture Notes in Computer Science 1807*, pages 557–572, 2000.

[36] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002), Lecture Notes in Computer Science 2482*, pages 54–68, 2002.

[37] T. Dierks and C. Allen. The TLS Protocol, version 1.0. `http://www.ietf.org/rfc/rfc2246.txt`, 1999.

[38] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[39] R. Dingledine, M. J. Freedman, D. Hopwood, and D. Molnar. A Reputation System to Increase MIX-Net Reliability. In I. S. Moskowitz, editor, *Fourth International Workshop on Information Hiding (IH'01), Lecture Notes in Computer Science, 2137*, pages 126–141, 2001.

[40] R. Dingledine, M. J. Freedman, and D. Molnar. The Free Haven Project: Distributed Anonymous Storage Service. In H. Federrath, editor, *DIAU'00, Lecture Notes in Computer Science 2009*, pages 67–95, 2000.

[41] R. Dingledine and P. Syverson. Reliable MIX Cascade Networks through Reputation. In *Financial Cryptography*, 2002.

[42] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *23th Symposium on the Theory of Computation (STOC)*, pages 542–552, 1991.

[43] S. Dolev and R. Ostrovsky. XOR-trees for Efficient Anonymous Multicast and Reception. *ACM Transactions on Information and System Security (TISSEC)*, 3(2):63–84, 2000.

[44] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In *ACM CCS*, pages 42–51, 2003.

[45] P. Erdös, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics*, 51(1-2):75–89, 1985.

[46] L. Eschenauer and V. D. Gligor. A Key-Management Scheme for Distributed Sensor Networks. In *ACM CCS*, pages 41–47, 2002.

[47] U. Feige. Alternative Models for Zero-Knowledge Interactive Proofs. Ph.D. Dissertation, Dept. of Computer Science and Applied Mathematics, Weizmann Institute of Science, 1990.

[48] U. Feige, D. Lapidot, and A. Shamir. Multiple Non-Interactive Zero-Knowledge Proofs Based on a Single Random String. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 308–317, 1990.

[49] U. Feige, D. Lapidot, and A. Shamir. Multiple Non-Interactive Zero-Knowledge Proofs Under General Assumptions. *SIAM Journal on Computing*, 29(1):1–28, 1999.

[50] H. Feistel. Cryptography and Computer Privacy. *Scientific American*, 228(5):15–23, 1973.

[51] P. Feldman. A Practical Scheme for Non-interactive Verifiable Secret Sharing. In *Symposium on Foundations of Computer Science (FOCS)*, pages 427–437, 1987.

[52] A. Fiat and M. Naor. Broadcast Encryption. In D. R. Stinson, editor, *CRYPTO'93, Lecture Notes in Computer Science 773*, pages 480–491, 1994.

[53] M. J. Freedman and R. Morris. Tarzan: A Peer-to-Peer Anonymizing Network Layer. In V. Atluri, editor, *9th ACM Conference on Computer and Communications Security (CCS'02)*, 2002.

[54] O. Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, 2001.

[55] O. Goldreich and L. A. Levin. A Hard-Core Predicate for all One-Way Functions. In *Symposium on the Theory of Computation (STOC)*, pages 25–32, 1989.

[56] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interaction Proof Systems. *SIAM Journal on Computing*, 18(4):186–208, 1989.

[57] C. Gülcü and G. Tsudik. Mixing E-mail With Babel. In *Network and Distributed Security Symposium - NDSS '96*, pages 2–16, 1996.

[58] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, 2000.

[59] P. Gupta and P. R. Kumar. Internets in the Sky: The Capacity of Three Dimensional Wireless Networks. *Communications in Information and Systems*, 1(1):39–49, 2001.

[60] V. Gupta, S. Gupta, and D. Stebila. Performance Analysis of Elliptic Curve Cryptography for SSL. In *First ACM Workshop on Wireless Security (WiSe)*, pages 87–94, 2002.

[61] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[62] C. Hedrick. Routing Information Protocol. `http://www.ietf.org/rfc/rfc1058.txt`, 1988.

[63] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive Secret Sharing or: How to Cope with Perpetual Leakage. extended abstract, IBM T.J. Watson Research Center, November 1995.

[64] Y.-C. Hu, D. B. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad Hoc Networks. In *Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02)*, 2002.

[65] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks. In *ACM MOBICOM*, pages 12–23, 2002.

[66] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks. In *IEEE INFOCOM*, 2003.

[67] Y.-C. Hu, A. Perrig, and D. B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. In *ACM WiSe'03 in conjunction with MOBICOM'03*, pages 30–40, 2003.

[68] J. Hubaux, L. Buttyan, and S. Čapkun. The Quest for Security in Mobile Ad Hoc Networks. In *MobiHOC*, 2001.

[69] E. Hughes. A Cypherpunk's Manifesto. `http://www.activism.net/cypherpunk/manifesto.html`.

[70] M. Jakobsson. A Practical MIX. In *EUROCRYPT'98, Lecture Notes in Computer Science 1403*, pages 448–461, 1998.

[71] M. Jakobsson. Flash Mixing. In *Principles of Distributed Computing - PODC '99*, 1999.

[72] M. Jakobsson, A. Juels, and R. Rivest. Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking. In D. Boneh, editor, *USENIX Security Symposium*, pages 339–353, 2002.

[73] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.

[74] D. B. Johnson and D. A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), April 2003.

[75] V. Kawadia and P. R. Kumar. Power Control and Clustering in Ad Hoc Networks. In *IEEE INFOCOM*, pages 459–469, 2003.

[76] D. Kesdogan, J. Egner, and R. Buschkes. Stop-and-go MIXes Providing Probabilistic Security in an Open System. *Second International Workshop on Information Hiding (IH'98), Lecture Notes in Computer Science 1525*, pages 83–98, 1998.

[77] J. Kong, S. Das, E. Tsai, and M. Gerla. ESCORT: A Decentralized and Localized Access Control System for Mobile Wireless Access to Secured Domains. In *ACM WiSe'03 in conjunction with MOBICOM'03*, pages 51–60, 2003.

[78] J. Kong and X. Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. In *ACM MOBIHOC'03*, pages 291–302, 2003.

[79] J. Kong, X. Hong, and M. Gerla. A New Set of Passive Routing Attacks in Mobile Ad Hoc Networks. In *IEEE MILCOM*, 2003.

[80] J. Kong, M. Mirza, J. Shu, C. Yoedhana, M. Gerla, and S. Lu. Random Flow Network Modeling and Simulations for DDoS Attack Mitigation. In *International Conference on Communications (ICC)*, 2003.

[81] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks. In *IEEE ICNP'01*, pages 251–260, 2001.

[82] H. Krawczyk. The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?). In J. Kilian, editor, *CRYPTO'01, Lecture Notes in Computer Science 2139*, pages 310–331, 2001.

[83] R. Kumar, S. Rajagopalan, and A. Sahai. Coding Constructions for Blacklisting Problems without Computational Assumptions. In M. J. Wiener, editor, *CRYPTO'99, Lecture Notes in Computer Science 1666*, pages 609–623, 1999.

[84] V. Lakshmi and D. P. Agrawal. An Optimized Inter-router Authentication Scheme for Ad Hoc Networks. In *International Conference of Wireless Communications*, pages 129–146, 2001.

[85] D. Lapidot and A. Shamir. Publicly Verifiable Non-Interactive Zero-Knowledge Proofs. In A. J. Menezes and S. A. Vanstone, editors, *CRYPTO'90, Lecture Notes in Computer Science 537*, pages 353–365, 1990.

[86] S.-J. Lee and M. Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks. In *IEEE International Conference on Communications (ICC)*, pages 3201–3205, 2001.

[87] A. K. Lenstra and E. R. Verheul. Selecting Cryptographic Key Sizes. In *Public Key Cryptography*, pages 446–465, 2000.

[88] J. Li, C. Blake, D. D. Couto, H. I. Lee, and R. Morris. Capacity of Ad Hoc Wireless Networks. In *ACM MOBICOM*, pages 61–69, 2001.

[89] D. Liu and P. Ning. Establishing Pairwise Keys in Distributed Sensor Networks. In *ACM CCS*, pages 52–61, 2003.

[90] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands, North-Holland, 1988.

[91] M. K. Marina and S. R. Das. Ad Hoc On-demand Multipath Distance Vector Routing. In *IEEE ICNP*, pages 14–23, 2001.

[92] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *ACM MOBICOM*, 2000.

[93] C. J. Mitchell and F. C. Piper. Key Storage in Secure Networks. *Discrete Applied Mathematics*, 21(3):215–228, 1988.

[94] M. Mitomo and K. Kurosawa. Attack for Flash MIX. In *ASIACRYPT'00, Lecture Notes in Computer Science 1976*, pages 192–204, 2000.

[95] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster Protocol — Version 2. `http://www.abditum.com/mixmaster-spec.txt`, July 2003.

[96] J. Moy. OSPF Version 2. `http://www.ietf.org/rfc/rfc1131.txt`, 1991.

[97] M. Narasimha, G. Tsudik, and J. H. Yi. On the Utility of Distributed Cryptography in P2P and MANETs: the Case of Membership Control. In *IEEE International Conference on Network Protocols (ICNP)*, pages 336–345, 2003.

[98] A. Nasipuri and S. R. Das. On Demand Multipath Routing for Mobile Ad Hoc Networks. In *IEEE International Conference on Computer Communication and Networks (ICCCN)*, 1999.

[99] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *ACM MOBICOM*, pages 151–162, 1999.

[100] W. Ogata, K. Kurosawa, K. Sako, and K. Takatani. Fault Tolerant Anonymous Channel. In *First International Conference of Information and Communications Security (ICICS), Lecture Notes in Computer Science 1334*, pages 440–444, 1997.

[101] R. Ogier, M. Lewis, and F. Templin. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). `http://www.ietf.org/internet-drafts/draft-ietf-manet-tbrpf-07.txt`, March 2003.

[102] M. Ohkubo and M. Abe. A Length-Invariant Hybrid MIX. In *ASIACRYPT'00, Lecture Notes in Computer Science 1976*, pages 178–191, 2000.

[103] P. Papadimitratos and Z. J. Haas. Secure Data Transmission in Mobile Ad Hoc Networks. In *Second ACM Workshop on Wireless Security (WiSe)*, pages 41–50, 2003.

[104] P. Papadimitratos, Z. J. Haas, and E. G. Sirer. Path Set Selection in Mobile Ad Hoc Networks. In *ACM MOBIHOC*, pages 160–170, 2002.

[105] C. Park, K. Itoh, and K. Kurosawa. Efficient Anonymous Channel and All/Nothing Election Scheme. In T. Helleseth, editor, *EUROCRYPT'93, Lecture Notes in Computer Science 765*, pages 248–259, 1993.

[106] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi. On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks. In *ACM MOBIHOC*, pages 3–10, 2000.

[107] C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM*, pages 234–244, 1994.

[108] C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *IEEE WMCSA'99*, pages 90–100, 1999.

[109] A. Perrig, R. Canetti, B. Briscoe, D. Tygar, and D. Song. TESLA: Multicast Source Authentication Transform. draft-irtf-smug-tesla-00.txt, June 2001.

[110] A. Perrig, R. Canetti, D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5(2):2–13, 2002.

[111] A. Pfitzmann and M. Köhntopp. Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In H. Federrath, editor, *DIAU'00, Lecture Notes in Computer Science 2009*, pages 1–9, 2000.

[112] A. Pfitzmann, B. Pfitzmann, and M. Waidner. ISDNMixes: Untraceable Communication with Very Small Bandwidth Overhead. In *GI/ITG Conference: Communication in Distributed Systems*, pages 451–463, 1991.

[113] A. Pfitzmann and M. Waidner. Networks Without User Observability: Design Options. In F. Pichler, editor, *EUROCRYPT'85, Lecture Notes in Computer Science 219*, pages 245–253, 1986.

[114] B. Pfitzmann and A. Pfitzmann. How to Break the Direct RSA-Implementation of MIXes. In J.-J. Quisquater and J. Vandewalle, editors, *EUROCRYPT'89, Lecture Notes in Computer Science 434*, pages 373–381, 1990.

[115] J. Postel. Internet Protocol. `http://www.ietf.org/rfc/rfc791.txt`, 1981.

[116] M. O. Rabin. Digital Signatures and Public Key Functions as Intractable as Factorization. Technical Report TM-212, Laboratory of Computer Science, Massachusett Institute of Technology, 1979.

[117] C. Rackoff and D. R. Simon. Cryptographic defense against traffic analysis. In *Symposium on the Theory of Computation (STOC)*, pages 672–681, 1993.

[118] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communications*, 16(4), 1998.

[119] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[120] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). `http://www.ietf.org/rfc/rfc1771.txt`, 1995.

[121] R. L. Rivest, A. Shamir, and L. M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *CACM*, 21(2):120–126, 1978.

[122] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *Symposium on Foundations of Computer Science (FOCS)*, pages 543–553, 1999.

[123] K. Sako and J. Kilian. Receipt-Free MIX-Type Voting Scheme - A Practical Solution to the Implementation of a Voting Booth. In L. C. Guillou and J.-J. Quisquater, editors, *EUROCRYPT'95, Lecture Notes in Computer Science 921*, pages 393–403, 1995.

[124] A. D. Santis, G. D. Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust Non-interactive Zero Knowledge. In J. Kilian, editor, *CRYPTO'01, Lecture Notes in Computer Science*, pages 566–598, 2001.

[125] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. Royer. A Secure Routing Protocol for Ad Hoc Networks. In *10th International Conference on Network Protocols (IEEE ICNP'02)*, 2002.

[126] N. Saxena, G. Tsudik, and J. H. Yi. Admission Control in Peer-to-Peer: Design and Performance Evaluation. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 104–114, 2003.

[127] Scalable Network Technologies (SNT). QualNet. `http://www.qualnet.com/`.

[128] B. Schoenmakers. A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting. In *CRYPTO*, pages 148–164, 1999.

[129] A. Serjantov and G. Danezis. Towards an Information Theoretic Metric for Anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002), Lecture Notes in Computer Science 2482*, pages 41–53, 2002.

[130] A. Serjantov, R. Dingledine, and P. F. Syverson. From a Trickle to a Flood: Active Attacks on Several Mix Types. In F. A. P. Petitcolas, editor, *Fifth International Workshop on Information Hiding (IH'02), Lecture Notes in Computer Science, 2578*, pages 36–52, 2002.

[131] A. Shamir. On the Generation of Cryptographically Strong Pseudo-Random Sequences. In S. Even and O. Kariv, editors, *International Colloquium on Automata, Languages and Programming (ICALP'81), Lecture Notes in Computer Science 115*, pages 544–550, 1981.

[132] C. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[133] C. Shields and B. N. Levine. A protocol for anonymous communication over the Internet. In *ACM Conference on Computer and Communications Security (CCS 2000)*, pages 33–42, 2000.

[134] *Journal on Selected Areas in Communications (J-SAC), Special issue on Software Radios*, volume 17-4. IEEE, April 1999.

[135] M. Stadler. Publicly Verifiable Secret Sharing. In *EUROCRYPT*, pages 190–199, 1996.

[136] D. R. Stinson, R. Wei, and L. Zhu. Some New Bounds for Cover-Free Families. *Journal of Combinatorics Theory*, A(90):224–234, 2000.

[137] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory. GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems. `http://pcl.cs.ucla.edu/projects/glomosim/`.

[138] S. Čapkun and J.-P. Hubaux. BISS: Building Secure Routing out of an Incomplete Set of Security Association. In *Second ACM Workshop on Wireless Security (WiSe)*, pages 21–30, 2003.

[139] S. Čapkun, J.-P. Hubaux, and L. Buttyan. Mobility Helps Security in Ad Hoc Networks. In *ACM MOBIHOC*, pages 46–56, 2003.

[140] G. S. Vernam. Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications. *Journal American Institute of Electrical Engineers*, XLV:109–115, 1926.

[141] M. Waldman, A. Rubin, and L. Cranor. Publius: A Robust, Tamper-evident, Censorship-resistant and Source-anonymous Web Publishing System. In *the 9th USENIX Security Symposium*, pages 59–72, 2000.

[142] WAP Forum. Wireless Transport Layer Security Specification. `http://www1.wapforum.org/tech/documents/WAP-261-WTLS-20010406-a.pdf`.

[143] M. Wright, M. Adler, B. N. Levine, and C. Shields. An Analysis of the Degradation of Anonymous Protocols. In *Network and Distributed Security Symposium - NDSS '02*, 2002.

[144] H. Yang and S. Lu. Self-Organized Network Layer Security in Mobile Ad Hoc Networks. In *First ACM Workshop on Wireless Security (WiSe)*, pages 11–20, 2002.

[145] A. C.-C. Yao. Theory and Applications of Trapdoor Functions (Extended Abstract). In *Symposium on Foundations of Computer Science (FOCS)*, pages 80–91, 1982.

[146] Zero-Knowledge. Freedom System. `http://www.freedom.net/`.

[147] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *ACM MOBICOM*, 2000.

[148] L. Zhou. Distributed Trust in Ad Hoc Networks. *Report on a Working Session on Security in Wireless Ad Hoc Networks, ACM Mobile Computing and Communications Review*, 6(4), 2002.

[149] L. Zhou and Z. J. Haas. Securing Ad Hoc Networks. *IEEE Networks*, 13(6):24–30, 1999.

[150] S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach. In *IEEE International Conference on Network Protocols (ICNP'03)*, 2003.

INDEX