Catching "Moles" in Sensor Networks

Fan Ye, Hao Yang, Zhen Liu
{fanye,haoyang,zhenl}@us.ibm.com
IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532

Abstract

False data injection is a severe attack that compromised sensor nodes ("moles"[1]) can launch. These moles inject large amount of bogus traffic that can lead to application failures and exhausted network resources. Existing sensor network security proposals only passively mitigate the damage by filtering injected packets; they do not provide active means for fight back. This paper studies how to locate such moles within the framework of packet marking, when forwarding moles collude with source moles to manipulate the marks. Existing Internet traceback mechanisms do not assume compromised forwarding nodes and are easily defeated by manipulated marks. We propose a Probabilistic Nested Marking (PNM) scheme that is secure against such colluding attacks. No matter how colluding moles manipulate the marks, PNM can always locate them one by one. We prove that nested marking is both sufficient and necessary to resist colluding attacks. PNM also has fast-traceback: within about 50 packets, it can track down a mole up to 20 hops away from the sink. This virtually prevents any effective data injection attack: moles will be caught before they have injected any meaningful amount of bogus traffic.

Keywords: Traceback, Sensor Networks, Colluding Attacks, Packet Marking

1 Introduction

Many wireless sensor networks are expected to work in a possibly adverse or even hostile environment. Due to their unattended operations, it is easy for an adversary to physically pick up and compromise sensor nodes, obtaining their stored data including secret keys. These compromised "moles" can launch various types of attacks, an important one of which is *false data injection* [12, 14]. One single mole can inject large amounts of bogus traffic to flood the sink, leading to application failures and wasting energy and bandwidth resources along the forwarding path. Recent research [12, 14, 11] has proposed a number of schemes to detect and drop such bogus messages en-route. However, they are all *passive* in that they only mitigate the damage of attacks. They do not provide active means for fight-back.

In this paper we study a crucial problem toward such active fight-back, that is, how to locate moles in sensor networks. Knowing their locations, we can isolate or remove them from the network, thus eradicating the root cause of the attack. Locating moles presents great research challenges. First, different from the Internet where routers are better protected and relatively trusted than end hosts, all sensor nodes are equally accessible by the adversary and uniformly un-protected. Any forwarding node may be compromised; there is no relatively trusted routing infrastructure that we can leverage. Second, the moles can collude. They can not only share their secret keys, but also manipulate packets in a coordinated manner to cover up their traces. Such manipulation attacks are far more sophisticated than simply increasing the amount of bogus traffic. Existing IP traceback schemes for the Internet [9, 8, 10, 4] do not consider such compromised forwarding nodes and become ineffective under such colluding attacks.

We propose a Probabilistic Nested Marking (PNM) scheme to locate colluding moles in false data injection attacks. We use packet marking [8] to discover the true origin of packets: A node marks its identity in the packets it forwards. By collecting such marks, the sink can infer the route, thus the origin location of the traffic. Although packet marking has been well explored in the Internet [8, 10, 4], its applicability against colluding sensor moles, however, has never been studied. Existing marking schemes for IP traceback can be easily defeated by an intermediate forwarding mole, which tampers the marks to hide the true locations of the source and itself, or even lead the sink to track to innocent nodes.

PNM achieves secure and efficient traceback against colluding moles using two techniques, namely *nested marking* and *probabilistic marking*. Nested marking supports single-

[1] "Moles" are spies who operate from within an organization, especially agents operating against their own governments. We use it to refer to compromised sensor nodes.

27th International Conference on Distributed Computing Systems (ICDCS'07)
0-7695-2837-3/07 $20.00 © 2007 IEEE

packet traceback. Each forwarding node marks packets in a nested fashion such that its mark protects the marks from all previous forwarding nodes. This ensures that no matter how a colluding mole manipulates the marks, it either reveals the source' location, or that of its own. Probabilistic marking reduces the per-packet marking overhead to suit the resource-constrained sensors. Each node leaves a mark with certain probability, thus a packet carries only a few marks. Different from Internet marking schemes where a new mark may replace an existing one, in PNM new marks are simply appended to the packet.

Using formal security analysis, we prove that nested marking is not only sufficient but also necessary for tracing to a mole's one-hop neighborhood. Moreover, we demonstrate the effectiveness and efficiency of PNM through analytical and empirical evaluations. PNM provides fast traceback: within about 50 packets, the sink can locate a mole up to 20 hops away. It virtually prevents moles from launching effective data injection attacks, as they will be caught before they can inject a meaningful amount of attack traffic. To the best of our knowledge, ours is the first work that thoroughly investigate the applicability of marking in sensor networks, and the first that defeats the cover-up of colluding moles.

We make several contributions in this paper. First, we point out the need for *proactive* security against moles in sensor networks. We also examine, within the packet marking framework, various colluding attacks that the moles can launch. Second, we thoroughly investigate the design space of packet marking and show that nested marking is both sufficient and necessary: if any portions of the previous nodes' marks are not protected (as in many seemingly natural designs), there exist attacks where a colluding mole can either hide the locations of the source and itself, or trick the sink to trace to innocent nodes. Third, we show that a straightforward probabilistic extension to nested marking is subject to one colluding attack of selective dropping. To defeat this attack, we propose an effective probabilistic nested marking scheme where the IDs of marking nodes are anonymized.

The rest of the paper is organized as follows. Section 2 presents the network and threat models. Section 3 demonstrates the insecurity of existing IP traceback schemes under colluding attacks. Section 4 presents the basic nested marking and probabilistic marking designs in PNM. Section 5 analyzes the security of PNM and proves why nested marking is both sufficient and necessary. Section 6 evaluates the performance of PNM and Section 7 discusses a number of practical issues. Section 8 compares PNM with the related work and Section 9 concludes the paper.

2 Models and Assumptions

2.1 System Model

We consider a static sensor network where sensor nodes do not move once deployed. These nodes sense the nearby environment and produce reports about interested events, which contain the time, location and description (e.g., sensor readings) of the events. The reports are forwarded to a sink by intermediate nodes through multi-hop wireless channels. The sink is a powerful machine with sufficient computing and energy resources.

The sensor nodes are resource-constrained and have limited computational power, storage capacity and energy supply. For example, the Mica2 motes [1] are battery powered and equipped with only a 4MHz processor and 256K memory. While public-key cryptography can be implemented in such low-end devices, it is too expensive in energy consumption. Thus we only consider efficient symmetric cryptography (e.g., secure hash functions) in our design.

We assume the routing is relatively stable. Routes do not change frequently in short time periods. When routes are stable, each node has only one next hop neighbor in its forwarding path and forwards all packets to the sink through this neighbor. This is consistent in tree-based routing protocols [6] or geographical forwarding [5].

We also assume that each sensor node has a unique ID and shares a unique secret key with the sink. The ID and key can be pre-loaded into a node before it is deployed. The sink can maintain a lookup table for all node IDs and keys. While nodes may establish other keys for purposes such as neighbor authentication, PNM does not require such keys to work.

2.2 Threat Model and Attack Taxonomy

The adversary may compromise sensor nodes through physical capture or software bugs, thus gaining full control of them. He has access to all the stored information, including secret keys, and can re-program them to behave in a malicious manner. We call such compromised nodes "moles". Moles may coordinate to maximize the damage. The sink is usually well protected. Although possible, we do not considered compromised sinks in this paper.

The context for traceback is the threat of *false data injection*. As illustrated in Figure 1, one mole S acts as a source and injects large amount of bogus sensing reports into the network. Such reports not only disrupt the user application but also waste network resources (e.g., energy, bandwidth) spent in forwarding them [12]. Traceback is the first step toward active defense. It allows the sink to identify the true origins of reports. The sink can then dispatch task forces to such locations remove moles physically, or notify their neighbors not to forward traffic from them. We leave the exact mechanism as future work and focus on traceback in this paper.

The challenge for an effective marking scheme is, a colluding mole X along the forwarding path may tamper the marks arbitrarily (see Figure 1). It can hide both its location and the source mole's location, or even trick the sink trace to innocent nodes. Hiding their locations allows continu-

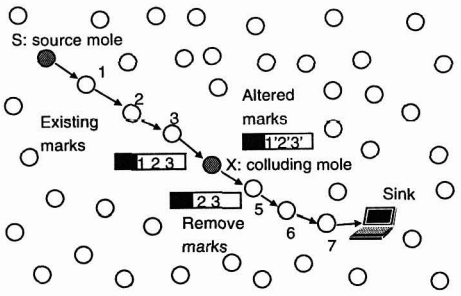

Figure 1. Moles S and X work together to cover their traces for injecting attack traffic. S injects bogus reports. X receives a packet with nodes $1, 2, 3$'s marks. X may manipulate the marks in various ways, such as altering these marks to $1', 2', 3'$, or remove the mark of node 1. The moles' goal is to hide their locations, or lead the sink trace to innocent nodes.

ous injection without being punished. This is needed for the injection to cause significant damage. Leaking any of their locations will lead to punishment such as network isolation or physical removal. Tricking the sink trace to innocent nodes is extra bonus: the sink may punish these nodes, thus denying legitimate resource and service to itself.

We present a taxonomy of colluding attacks against marking-based traceback by two colluding moles, S that injects bogus reports, and X on the forwarding path.

1) *No-Mark Attacks*: A mole may not mark the report at all. 2) *Mark Insertion Attacks*: Both the source mole and the forwarding mole may insert one or many faked marks into the reports. 3) *Mark Removal Attacks*: A forwarding mole may remove existing marks left by upstream nodes in the reports. 4) *Mark Re-ordering Attacks*: A forwarding mole may re-order existing marks in the reports. 5) *Mark Altering Attacks*: A forwarding mole may alter existing marks in the reports and make them invalid. 6) *Selective Dropping Attacks*: A forwarding mole may selectively drop those packets that, if received by the sink, would lead the traceback to them[2]. 7) *Identity Swapping Attacks*: S and X may know each other's key and impersonate each other.

For example, Figure 1 shows a chain of 7 forwarding nodes between a source mole S and the sink. Node X is the colluding mole. It receives V_3's message, which contains 3 valid marks $1, 2, 3$, left by nodes V_1, V_2, V_3. It may alter them to $1', 2', 3'$, making them invalid, thus the sink rejects these marks. It may remove mark 1 and leave only $2, 3$, thus the traceback stops at innocent node V_1.

2.3 Notations

To aid the presentation, we use the following notations. A source mole $\mathbf{S}$ injects bogus reports that conform to the legitimate format. Each report $\mathbf{M}$ contains an event E, location L and timestamp T (i.e., $M = E|L|T$, where "|" denotes concatenation). Bogus reports cannot all contain exactly the same content, otherwise they are considered redundant and be dropped by legitimate forwarding nodes. M is forwarded over a chain of n intermediate nodes $\{\mathbf{V_i}\}(i = 1, ..., n)$ to the sink.

Each node V_i has a unique ID i and shares a unique key $\mathbf{k_i}$ with the sink. It can use its key to generate a Message Authentication Code (MAC) for the packets it generates or forwards, using an efficient and secure keyed hash function $H_k(\cdot)$, where k is the key. Specifically, V_i adds a mark $\mathbf{m_i}$ to the message it receives from previous hop V_{i-1} to construct its own message $\mathbf{M_i}$. m_i may include V_i's ID i and MAC MAC_i. V_i then sends M_i to the next hop V_{i+1}.

Forwarding node V_x $(1 \leq x \leq n)$ is a colluding mole and we denote it $\mathbf{X}$ can manipulate the messages it receives from V_{x-1} in arbitrary manner, then pass it to V_{x+1}. It can use any one or a combination of the attacks in Section 2.2 to disrupt the traceback.

3 Internet Marking Schemes Not Applicable

A number of marking schemes [3, 8, 10, 4] have been proposed for IP traceback. They assume that the attacker compromises many end hosts, but usually not Internet routers. Routers simply mark the packets with their IP addresses in plain text without any security protection. Clearly, they cannot be directly applied in sensor networks where a forwarding mole can arbitrarily forge such marks. Nevertheless, the Authenticated Marking Scheme (AMS) [10] has considered compromised routers and cryptographically protects the marks. However, even AMS cannot withstand many colluding attacks from only two moles. Our purpose is not to criticize, but rather, illustrate why we need something different in sensor network context.

AMS protects the marks using a secure hash function. Each node shares a unique secret key with the sink. Upon receiving a packet, a forwarding node V_i probabilistically marks it with $H_{k_i}(IP_s|IP_d|i)$, where IP_s, IP_d are source and destination IP addresses[3]. In the original AMS, a packet carries at most one mark (due to the limits of available bits in the IP header). We extend it such that a packet can carry multiple marks, one from each forwarding node as $H_{k_i}(S|i)$ (Destination ID is removed as the sink is well-known in sensor network context). Consider the example shown in Figure 1, where S and X are two colluding moles. This extended AMS fails under mark removal, mark re-order, mark altering and selective drop attacks. For example, if mole X removes all marks from S and node 1, the sink will trace back to innocent node 2.

Extended AMS fails because the mark added by a node does not protect marks left by previous nodes. Each mark

[2]We do not consider the case where a forwarding mole drops all bogus traffic. In that case the sink cannot receive any such reports or marks, thus marking schemes are not applicable.

[3]This is one marking method as suggested by the authors in [10].

can be individually manipulated without affecting the validity of other marks. In the following, our basic nested marking establishes a binding between each mark and all previous marks. We will also show that a probabilistic marking requires an additional feature, anonymity of IDs, to defeat selective dropping attacks.

4 PNM Design

PNM can locate colluding moles, in the context of false data injection attacks, within the precision of a *single suspected neighborhood*. This includes one node and its one-hop neighbors, and there must be at least one mole among these nodes. PNM consists of two novel techniques, namely *nested marking* and *probabilistic nested marking*. Nested marking is the basic mechanism. It ensures that the sink can trace back to one mole at a time, using only one packet. However, it has a drawback of large message overhead since each forwarding node needs to place a mark on the packet. In large sensor networks this is not efficient.

Subsequently, we use probabilistic marking to spread the message overhead over multiple packets. Each forwarding node places a mark with certain probability. Thus a packet carries only a few marks and per-packet overhead is greatly reduced. This trades off detection power for less message overhead. The sink may need multiple packets to identify the moles, which is reasonable as long as the moles are identified before they cause significant damage.

4.1 Basic Nested Marking

Packet Marking: Each forwarding node V_i appends to the packet its ID i and a secure MAC using the secret key k_i it shares with the sink. The MAC protects the *entire* message it receives from V_{i-1}. That is, $MAC_i = H_{k_i}(M_{i-1}|i)$. As an example (see Figure 1), the messages sent by neighboring nodes are:

$$
\begin{aligned}
&S -> V_1 : M \\
&V_1 -> V_2 : M_1 = M|1|H_{k_1}(M|1) \\
&V_2 -> V_3 : M_2 = M_1|2|H_{k_2}(M_1|2) \\
&... \\
&V_i -> V_{i+1} : M_i = M_{i-1}|i|H_{k_i}(M_{i-1}|i)
\end{aligned}
$$

At each hop, the ID i indicates node i's presence on the route, the MAC $H_{k_i}(M_{i-1}|i)$ proves to the sink it is indeed node i that sends message M_i, and what the node receives was M_{i-1}. We can see that the MAC added by V_i protects not only its own ID but the entire message from the previous hop. This is where the name of *nested marking* comes from.

Due to the nested marking, any tampering with the previous IDs, or MACs, or their order, will make the MAC invalid. In Section 5, we will use formal security analysis to show that nested marking is *sufficient and necessary* for secure traceback. That is, it can withstand all colluding attacks, but any simpler design cannot. In extended AMS only the original message M and V_i's ID are protected, but not the mark's binding to previous marks in M_{i-1}. That is why it fails when marks are individually manipulated.

Traceback: After receiving packet M_n, the sink verifies the nested marks backwards. It first retrieves the ID of the last hop n and uses the corresponding key k_n to verify the last MAC MAC_n. If MAC_n is correct, it retrieves the ID of the previous hop $n-1$ and verifies MAC_{n-1}. The sink continues this process until either it has verified all MACs as correct, or it finds an incorrect MAC_x. A mole (either source or forwarding) is located within the one-hop neighborhood of the node with the last verified MAC (including this node itself).

In the example shown in Figure 1, node X is a mole. If X alters the mark of node 1, marks from nodes 1, 2 and 3 will all become invalid. When X does not leave a mark or leaves an invalid mark, the traceback stops at node 5 and a mole (X) is among the one-hop neighbors of this stopping node; when X leaves a valid mark, the traceback stops at node X, in which case the stopping node itself is a mole.

4.2 Probabilistic Nested Marking

The basic idea of Probabilistic Nested Marking is to let each forwarding node mark the packet with a small probability p. Thus on a forwarding path of n nodes, on average a message carries np marks. The probability p can be tuned such that the overhead of np marks is acceptable.

An Incorrect Extension: Extending to a probabilistic marking may look straightforward at first glance. However, it turns out to be non-trivial. Simply letting each node mark with probability p (see the following) is vulnerable to selective dropping attacks that can lead the traceback to innocent nodes.

$$
\begin{aligned}
&S -> V_1 : M \\
&V_1 -> V_2 (with\ p) : M_1 = M|1|H_{k_1}(M|1) \\
&V_1 -> V_2 (with\ 1-p) : M_1 = M \\
&... \\
&V_i -> V_{i+1} (with\ p) : M_i = M_{i-1}|i|H_{k_i}(M_{i-1}|i) \\
&V_i -> V_{i+1} (with\ 1-p) : M_i = M_{i-1}
\end{aligned}
$$

Consider the example in Figure 1. Since the ID list is in plain text, the colluding mole X can see which of V_1, V_2, V_3 have marked the packet. It can drop all packets containing marks of V_1, and forward just those bearing marks from V_2, V_3. When the sink traces back, it will stop at V_2, whose one-hop neighborhood does not contain any mole. Actually, X can lead the traceback to any innocent node between itself and the source mole.

This attack works because in probabilistic marking, each packet carries only partial "samples" of nodes on the forwarding path. Due to the plain text ID, the mole can selectively pass certain "samples" so that the sink sees only a partial path ending at one of X's upstream nodes. It does not work in the basic nested marking, because every packet

carries marks constituting the complete path. There exists no partial "samples" for selective dropping.

We face a dilemma here. We do not want any node be able to tell who have marked the packet. This way the colluding mole cannot know which packets to drop. However, the sink still needs to find out who have left marks to verify them. In the following, we exploit the asymmetry of the sink, extra knowledge about all secret keys and sufficient computing resources, to solve the problem.

Probabilistic Nested Marking: Instead of using its real ID i, a legitimate node V_i uses an anonymous ID i' in the packet. The mapping from real ID i to anonymous ID i' depends on the secret k_i, known by only V_i and the sink. The colluding mole does not possess the knowledge of keys from uncompromised nodes, thus it cannot deduce the real ID from the anonymous one.

$$
\begin{aligned}
&S->V_1 : M \\
&V_1->V_2(with\ p) : M_1 = M|1'|H_{k_1}(M|1'), \\
&where\ 1' = H'_{k_1}(M|1) \\
&V_1->V_2(with\ 1-p) : M_1 = M \\
&... \\
&V_i->V_{i+1}(with\ p) : M_i = M_{i-1}|i'|H_{k_i}(M_{i-1}|i'), \\
&where\ i' = H'_{k_i}(M|i) \\
&V_i->V_{i+1}(with\ 1-p) : M_i = M_{i-1}
\end{aligned}
$$

In the above, $H'()$ is another secure one-way function that computes the anonymous ID. The anonymous ID i' is bound to M such that it changes for each distinct message V_i forwards[4]. This avoids a static mapping that can be accumulated over time by the attacker. Compared to the extended AMS, it has both nested marking and anonymous ID.

Mark Verification With the anonymous ID, the verification at the sink becomes different. It first needs to know the real ID to decide which secret key to use to verify the MAC. We exploit the abundant computing power at the sink to search for the real ID.

After receiving M_n from node V_n, the sink first computes all the anonymous IDs for every node in the network. Knowing M, it can build a table to map all IDs i to i'. By looking up i', it knows the real ID i. Then it can use the corresponding key k_i to verify the MAC. This way, it can verify all MACs one by one.

This search is feasible given the sink's computing power and the low data rate in sensor networks. For each distinct message M, it needs to compute a different table to do the lookup. Given that hash computation can be done at microsecond level (e.g., an Athlon 1.6G CPU can do 2.5 million hash per second[5]), building such a table for even a reasonably large network (a few thousand nodes) should take on the order of a few milliseconds. Thus the sink can verify several hundred or more packets per second. Because the sink receives from one sensor at a time, the incoming data rate is limited by the radio rate of sensors. Several hundred packets is already much higher than the current actual data rate on typical sensor hardware (e.g., 19.2kbps for Mica2 motes, around 50 packets per second[6]).

Traceback Locating moles becomes a two-step process. First the sink needs to reconstruct the route by collecting marks from a sufficient number of packets (the exact number will be analyzed in Section 6). Then it identifies which nodes have moles in their one-hop neighborhood. Due to space limit, We briefly explain the main idea of the algorithm that the sink uses to locate moles (pseudo code omitted).

The route can be reconstructed by finding the relative order of nodes (which is upstream to which) in the forwarding path. We use a matrix M to maintain the relative orders. The matrix is initially empty. When a correct MAC for a new node V_i is verified, one more row and one more column corresponding to V_i is added to the matrix. Whenever two consecutive MACs MAC_i, MAC_j within one packet are verified as correct, V_i should be upstream to V_j, and $M[i,j]$ records this relation (e.g., be set to 1) in the matrix. As more packets are received, the sink keeps updating this matrix. Given sufficient packets, the sink will be able to find out the upstream relation among all forwarding nodes, thus the complete route.

The sink may reconstruct two types of routes: those that do not have loops, or those have loops. The first type happens when moles use attacks other than identity swapping, the latter when moles swap their identities to leave marks. In the first case, locating moles is equivalent to finding the most upstream node. Because a source mole produces packets by itself, it does not receive packets from others and it can be the most upstream node. A forwarding mole may "appear" to be the most upstream, if it removes marks left by its upstream nodes. In either case, a source or forwarding mole, is within the one-hop neighborhood of the most upstream node.

The moles may use identity swapping to create loops (see Figure 2), thus there does not exist a "most" upstream node. A source mole S and a forwarding mole X may leave valid marks using the key of each other for some packets, and use their own keys for some other packets. The sink will find that S appears before X for some packets, and after X for other packets. It will also find that all nodes between S and X (including them) form a loop. For any two nodes U, V in such a loop, U appears both upstream and downstream to V.

[4]Remember that to avoid being considered as redundant copies and dropped, reports forged by the source mole have different content.

[5]These numbers are based on the measurement shown in http://www.azillionmonkeys.com/qed/hash.html

[6]http://mail.millennium.berkeley.edu/pipermail/tinyos-help/2003-June/001496.html

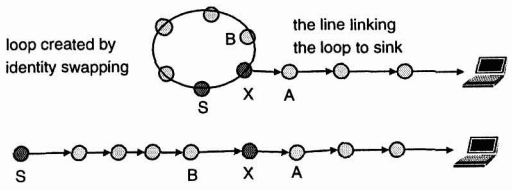

Figure 2. In this example, S and X use each other's key to leave valid marks for some packets. When the sink reconstructs the route, there is a loop containing all nodes between S and X (including them). The sink can still trace back to where the loop intersects the line and identify a mole within that neighborhood.

However, this anomaly can be easily identified: the sink can find the rest of the nodes form a line from the loop to itself. A mole is located within the one-hop neighborhood of the most upstream node in this line (i.e., where the loop intersects with the line). We will present detailed analysis in Section 5.3.

5 Security Analysis

We analyze the security strength of PNM and compare it to alternative marking schemes. Our analysis shows that nested marking is both precise and necessary: It can track down moles within one-hop neighborhood area despite colluding attacks, but any simpler design fails under certain attacks. The probabilistic nested marking can track down moles within one-hop neighborhood area *asymptotically* as the sink receives sufficient number of packets over time.

5.1 Security of Nested Marking

We first define two properties for any marking schemes, namely *one-hop precision* and *consecutive traceability*, and then prove that they are equivalent. Next we prove that our basic nested marking scheme is one-hop precise by showing its consecutive traceability.

Definition 5.1 (One-hop precision): *A marking scheme has one-hop precision in traceback if it can always trace to either the source node's or a colluding mole's one-hop neighborhood.*

Definition 5.2 (Consecutive Traceability): *Consider two consecutive legitimate nodes U and V on a forwarding path (i.e., V receives messages from U and then forwards them). With a consecutive traceable marking scheme, if the sink has traced to V, it can always further trace to U.*

Theorem 1 *A marking scheme is one-hop precise if and only if it is consecutive traceable.*

Proof: We first prove the sufficiency. Suppose that the traceback stops at a node V, which is the last node (in the reverse order of forwarding) that has a valid MAC. V cannot be a legitimate node that is not on the forwarding path, because such nodes will not generate MACs for messages they do not forward, while the attacker does not know their secret keys. Thus, V is a either a mole, or a legitimate node on the forwarding path. If V is a mole, the sufficiency holds. Next we consider the case where V is a legitimate node.

Let U be the previous hop of V, i.e., V receives messages from U. There are only two possibilities: either U is a mole (source or colluding) or U is a legitimate node. In the first case, the sufficiency holds because V is in the neighborhood of a mole U. On the other hand, by definition of consecutive traceability, the traceback will proceed to U and will not stop at V. Thus the second case cannot happen. This concludes the proof of sufficiency.

Next we prove the necessity. Suppose a marking scheme is not consecutive traceable. That is, there exists cases when the sink has traced to a legitimate node V, but it cannot proceed to the previous legitimate node U. Thus the traceback stops at V, not necessarily the neighborhood of the source or a colluding mole. By definition, such a marking scheme is not one-hop precise. □

The intuition behind Theorem 1 is as follows. There are only two categories of nodes on a forwarding path: a) moles and their immediate next hop, and b) legitimate nodes that have legitimate previous-hop neighbor. One-hop precision means the traceback stops at a node within the first category; consecutive traceability means the traceback cannot stop within the second category – thus it has to stop within the first category.

Theorem 2 *The nested marking scheme is consecutive traceable.*

Proof: Consider two consecutive legitimate forwarding nodes U and V. Let M_u be the message that U sends to V, and V sends $M_u|V|H_{k_v}(M_u|V)$ to the next hop.

Suppose the sink has traced to V. This means that it should have verified MAC'_v in a message $M'_u|V|MAC'_v$, and found that the recomputed MAC ($H_{k_v}(M'_u|V)$), is the same as the included MAC'_v. Because the attacker does not know k_v, MAC'_v must be the MAC_v generated by V. Thus M'_u and M_u must be the same; otherwise, the recomputed MAC would not match that produced by V.

Because M_u is sent by a legitimate node U, the last mark in M_u must carry a valid MAC from U. Therefore, by verifying this MAC, the sink can further trace to U. □

Corollary 5.1 *The nested marking scheme is one-hop precise.*

5.2 Necessity of Nested Marking

Theorem 3 *Any marking scheme that protects fewer fields than nested marking is not consecutive traceable.*

Proof: In the nested marking, a node's MAC protects both its own ID and the entire message it receives from the previous hop. Now consider an alternative marking scheme Γ, in which the MAC protects less fields. There must exist a node A, whose ID or MAC is not completely protected by all nodes after it; otherwise, Γ would become the nested marking scheme.

Let U be the last node that protects A's ID and MAC completely, and V be the next hop of U (See Figure 3). That is, there are some bits in A's mark not protected by V's MAC. Let us consider one mole downstream after V. The mole properly marks the report, and it alters the bits in A's mark not protected by V's MAC. In this case, the MACs of all nodes after V (including V) are correct, thus the sink can trace to V. However, because A's mark is tampered by the mole, U's MAC would appear invalid, thus the sink cannot further trace to U. In other words, Γ is not consecutive traceable. □

Corollary 5.2 *Any marking scheme that protects less fields than the nested marking is not one-hop precise.*

5.3 Security of PNM

Theorem 4 *The probabilistic nested marking is asymptotically one-hop precise if the routes are stable.*

Proof Sketch: Due to space limits, we only provide a sketch of the proof here. The full proof is available in [13]. Based on Theorem 1, it suffices to prove the asymptotic consecutive traceability of PNM. There are two possible cases in PNM: either the reconstructed path is loop-free or it has loops.

When there is no loop in the path, the proof is similar to that of Theorem 2, as long as we can ensure that for any two consecutive legitimate forwarding nodes, the sink receives at least one packet that is marked by both nodes. This holds asymptotically because each node independently marks the packets, and the moles cannot selectively drop such consecutively marked packets due to the use of anonymous IDs.

When the path has loops, we prove that the node at the intersection of a loop and a line must have moles within its one-hop neighborhood (including this node itself) by contradiction. In the illustrative example shown in Figure 2, node X joins the loop and the line, and 4 nodes (X, S, A, B) exists in its one-hop neighborhood. Suppose all of them are legitimate. Clearly, A is X's next-hop neighbor because some packets flow from X to A. Moreover, X must have also forwarded packets to at least one neighbor on the loop (either S or B). As such, X has at least two next-hop neighbors on its forwarding path. However, when routes are stable, any legitimate node should have only one next-hop neighbor for a given sink. Thus these 4 nodes cannot all be legitimate nodes and one of them must be a mole. □

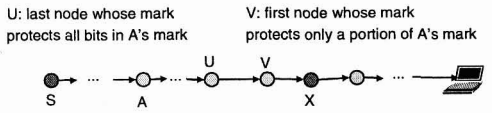

Figure 3. X alters the bits in A's mark that are not protected by V. Thus V's mark is still correct, but U's is not. The sink traces back to V, but cannot further trace to U.

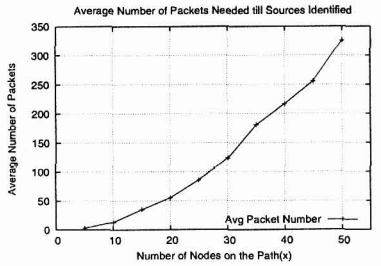

Figure 7. The average number of packets needed to unequivocally identify the source, as a function of total path length. 800 packets are received at the sink in each run.

6 Performance Evaluation

6.1 Analysis

We first analyze N, the number of packets needed for the sink to collect at least one mark from each of the forwarding nodes $V_1, ..., V_n$. We can compute the probability that this is achieved within L packets is (details in [13]):

$$P(N \leq L) = (1 - (1-p)^L)^n$$

Such a probability is illustrated in Figure 4, where the average number of marks a packet carries (that is, np) is fixed at 3. For a path containing 10 nodes, after receiving 13 packets, the sink has about 90% probability of having collected all marks. It takes 33 and 54 packets to achieve the 90% confidence for paths of 20, 30 hops respectively. The results show that after a relatively small number of packets, which have not wasted significant energy and bandwidth resources, the sink will have collected marks from all nodes.

6.2 Simulation Results

We use simulations to evaluate the performance of PNM from various aspects. In the simulations, we vary the number of forwarding nodes on a path, n, as $10, 20, 30$, and set the marking probability p such that a packet always carries 3 marks on average. For each parameter setting, the results reported are the average over 5000 runs.

We first evaluate how fast the sink can collect the marks. Figure 5 shows the percentage of nodes whose marks are collected by the sink using the first x packets. With a 10-hop path, on average the sink can collect marks from 9 nodes using only 7 packets. For 20-hop or 30-hop paths, it takes about 14 or 22 packets to collect marks from 90% of the nodes respectively. In other words, within a few dozen packets, the sink knows which are on the forwarding nodes.

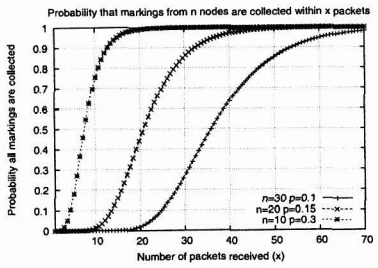

Figure 4. The probability that the sink collects marks from all n forwarding nodes with x packets.

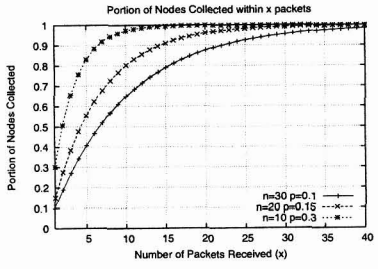

Figure 5. The average precentage of nodes whose marks are collected by the sink in the first x packets.

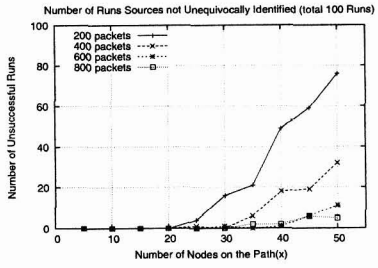

Figure 6. The number of runs, out of 100 simulations, in which the sink fails to unequivocally identify the source, as a function of total path length.

However, the sink needs much more packets to unequivocally reduce the candidate source set to moles only. To evaluate how many packets are needed for this purpose, we change the number of packets the sink receives as 200, 400, 600 and 800. For each traffic amount, we also try different path lengths ranging from 5 to 50. We record the number of times that the sink cannot unequivocally identify the source. Figure 6 illustrates the number of failed runs as a function of path length, for the 4 different traffic amount.

We can see that 200 packets are sufficient for up to 20-hops paths, as our algorithm can unequivocally identify the source in almost all runs. Moreover, 400 packets are enough for up to 30-hops paths. Only for very long paths (e.g., with 50 nodes), a relatively large number (e.g., 800) of packets are needed to reduce the failure frequency to less than 5%.

Finally, we take a closer look at the *average* number of packets the sink needs to unequivocally identify the source, over all cases where the sink can successfully do so. Figure 7 shows the results as a function of path length, where the traffic amount is fixed at 800 packets. We can see that for paths with less than 20 nodes, on average it takes about 55 packets to unequivocally identify the source. This roughly match the analytical results shown in Figure 4, where with 55 packets, the sink has over 99% probability of having collected marks from all the 20 fowarding nodes. Even for long paths such as 40 nodes, the sink can unequivocally identify the source after about 220 packets . The results demonstrate that PNM can prevent moles from launching effective false data injection attacks, because they will be located before inflicting sufficient damages to the network.

7 Discussions

In this section, we discuss several design issues in PNM.

Traceback Precision PNM can trace back to one-hop neighborhood of a mole, but not any specific nodes. This is because a mole can claim different identities in communicating with its neighbors. We can improve the traceback precision of PNM to a pair of neighboring nodes with additional neighbor authentication schemes, e.g., using pairwise keys. Such extensions are beyond the scope of this paper.

Mole Isolation PNM alone does not eliminate the root causes of false data injection attacks. It is expected to work together with some mole isolation mechanisms, so that once a mole is identified, it is either eradicated or quarantined in its local neighborhood. We will investigate the mole isolation mechanisms in the future.

Background Traffic For effective traceback, the sink must know which packets were generated by moles. However, legitimate traffic may co-exist with the attack traffic. The sink can identify suspicious packets in many ways, e.g., by verifying whether the reported events do exist, or checking traffic characteristic such as volume and route diversity. A thorough investigation on this issue is left for future work.

Impact of Routing Dynamics PNM assumes that the routes are stable during the traceback period. Given the fast traceback feature of PNM (e.g., about 10 seconds to locate a mole 40-hops away from the sink, using 300 packets), this assumption holds in most practical settings. Moreover, even if routing dynamics do occur during the traceback period, PNM can still locate the moles as long as the relative upstream relation among nodes remains the same.

Replay Attacks A source mole may seek to evade the PNM traceback by replaying past legitimate reports, which already contain a set of marks. Such replay attacks can be partially thwarted by duplicate message suppression at each forwarding node. A more effective solution can leverage packet sequence numbers that can be used one-time only; however, we do not elaborate the details here due to space limit.

Anonymous ID Mapping In PNM, the sink needs to map an anonymous ID to the node's real ID, which is currently done using an exhaustive search. This may not scale well for large networks or high radio rates. We note that if the sink knows the network topology[7], for each anonymous ID, it can limit the search within the one-hop neighbors of the previously verified node. As such, the search complexity is reduced to $O(d)$, where d is the degree of a node.

[7]In practice, one way to collect the network topology is to let each node report its neighbors to the sink after it is deployed.

8 Related Work

False data injection attack is an important security problem in sensor networks. Several en-route filtering schemes [12, 14, 11] have been proposed to drop the false data en-route before they reach the sink. However, these schemes only mitigate the threats. First, none of them can achieve perfect filtering. Second, filtering does not prevent moles from continuing to inject bogus reports. Even these reports are dropped after a few hops, they still waste the energy resource of legitimate nodes. Our traceback scheme complements the filtering ones by locating the moles. This makes it possible to physically remove or isolate such moles from the network, thus eradicating the root cause of the attacks.

A rich body of packet marking schemes [8, 10, 4] have been proposed for IP traceback in the Internet. They usually do not assume compromised forwarding nodes (i.e., routers) and are not designed to handle colluding moles on forwarding paths. As shown in Section 3, such moles can tamper the marks and trick the sink to trace to wrong nodes. Even the authenticated IP marking scheme [10] that considers compromised routers cannot withstand all colluding attacks. In contrast, our work is specifically designed to handle such colluding attacks from compromised forwarding nodes.

Besides packet marking, there are two more approaches for traceback, namely logging and notification. In logging schemes [9], each node stores the recently forwarded packets (or hash copies), and the sink can construct the path that a packet traverses by querying which nodes have forwarded it. In notification schemes [2], a forwarding node probabilistically notifies the sink of the packets they are forwarding (e.g., using ICMP messages). PNM differs from them in two aspects. First, it requires no control messages such as query/reply or notification. Securing these signaling mechanisms and preventing moles from abusing them is a challenging task. Second, it does not require a node to store any previously forwarded packets. This is particularly desirable for low-end sensors that have very limited storage capacity.

Techniques similar in spirit to nested marking have been used in other contexts, e.g., anonymous routing [7]. They address a problem opposite to ours, that is, how to prevent an attacker from tracing back to the sender or the receiver of a message. They also typically use public-key cryptography to construct the routing "onions." We study the problem of tracing back to the real sources of messages, and we do not require any public-key cryptography.

9 Conclusions and Future Work

False data injection attack has recently attracted much attention [12, 14, 11], and all existing solutions are passive in that they only mitigate the damage of attacks. Probabilistic Nested Marking is the first work that can locate moles despite colluding attacks. Combined with physical removal or network isolation, PNM can be used to actively fight back moles. We have formally proved its security against colluding moles and demonstrated its efficiency with analysis and simulation. PNM can track down a mole 20 hops away from the sink using only 50 packets. This essentially prevents effective data injection attacks, as moles will be caught before they can inflict any meaningful damages to the network.

We plan to continue our investigation along several directions. First, we will thoroughly evaluate the performance of PNM using real implementation on current sensor platforms. Second, we will study how to improve the traceback precision to specific nodes, and how to isolate the identified moles in the network. It is our conjecture that marking alone is insufficient and mechanisms such as neighbor authentication or collaborative monitoring may be needed. Finally, we will revisit the path reconstruction algorithm in the presence of multiple source moles.

References

[1] Xbow sensor networks. http://www.xbow.com/.

[2] S. Bellovin. ICMP traceback messages. In *Internet Draft: draft-bellovin-itrace-00.txt*, 2000.

[3] T. Doeppner, P. Klein, and A. Koyfman. Using router stamping to identify the source of IP packets. In *ACM CCS*, 2000.

[4] Q. Dong, M. Adler, S. Banerjee, and K. Hirata. Efficient probabilistic packet marking. In *IEEE ICNP*, 2005.

[5] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *ACM MOBICOM*, 2000.

[6] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TinyDB: An acquisitional query processing system for sensor networks. *ACM Tran. Database Systems*, 30(1), 2005.

[7] 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.

[8] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *ACM SIGCOMM*, 2000.

[9] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, S. Kent, and T. Strayer. Hash-based IP traceback. In *ACM SIGCOMM*, 2001.

[10] D. Song and A. Perrig. Advanced and authenticated marking schemes for IP traceback. In *IEEE INFOCOM*, 2001.

[11] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh. Toward resilient security in wireless sensor networks. In *ACM MOBIHOC*, 2005.

[12] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route filtering of injected false data in sensor networks. In *IEEE INFOCOM*, 2004.

[13] F. Ye, H. Yang, and Z. Liu. A secure and efficient traceback mechanism for sensor networks. Technical report, IBM Research, April 2007.

[14] S. Zhu, S. Setia, S. Jajodia, and P. Ning. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In *IEEE Symposium on Security and Privacy*, 2004.