Minimally Exposed Routing Information for Mobile Ad hoc Networks

Antonio Martin
Boston University
May 2006

***Abstract* - High security networks require all data, including routing information, to be encrypted between layers two and three of the OSI stack. As a result, nodes must decrypt every packet to examine if they are the intended receiver and can place a heavy decryption load on the nodes. We will examine possible means of reducing the packet decryption rates for the network as a whole without compromising security.**

I. INTRODUCTION

In IP packet based communication (TCP, UDP), even in high security modes like IPSec/HAIPE, to route this information over networks, an IP header must be exposed. This header includes the sending and receiving node's IP address and the protocol of encapsulation. This level of exposed data is necessary and acceptable for wired protocols since, in most cases, such packets are NATed, the true source and destinations are masked; the data inside the packet remains protected. This allows an eaves dropper to only distinguish that packets are traversing between two end points, at certain times and of certain size. It is possible that analysis of packet timing and size could lend some insight as to the protocols being used but this is beyond the scope of this document.

In a wireless network, exposing routing information reveals the topology of a network. It allows an observer to view communication types, paths/flows and node numbers. Wireless end nodes talk directly to other end nodes without the masking benefit of proxies such as NAT and/or other encapsulating means. Thus, exposed IP header information will uniquely identify the sender and to whom the packet is destined; in a high security situation, this is undesirable. To solve the problem, an entire packet, including the routing information in the header, must be encrypted. As a result, each receiver in range must decrypt every packet to examine if they are the destined target; this can be costly in terms of power consumption and processing requirements. Heavy network traffic can place a high demand on nodes and is of greater concern for mobile nodes where decryption requires MIPS and drains batteries.

Not only does this drain power from the radios in the network, but the network's performance is limited by the maximum processing capabilities of the slowest radio/node, since each and every packet must be decrypted and inspected. If a network produces more data then an individual radio can inspect, then said radio is, in effect, cut off from the network; this is of greater concern for soldier, hand held and legacy radios.

This problem applies to any wireless communication network but is compounded in an ad hoc network where an individual packet can be broadcasted multiple times. As a packet is sent, it can be re-broadcasted down a chain; thus a single packet might need to be decrypted by a very large number of the nodes in the network.

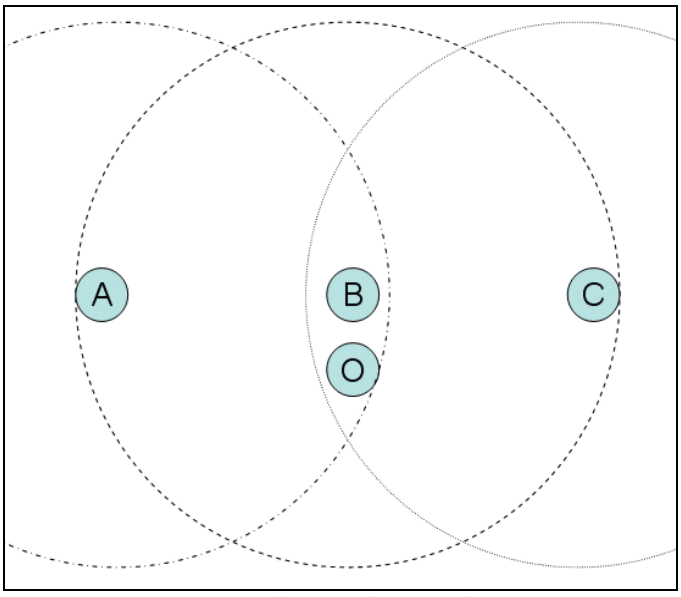

Figure 1: Basic topology used for teaching

Take for example the following scenario as shown in Figure 1. Node A wishes to send a packet to Node C with Node B acting an intermediary (A->B->C). In the neighborhood of Node B is Node O (observer). Nodes B and O can hear nodes A and C whiles nodes A and C cannot hear each other. When A sends a packet to Node B for routing to Node C, both Node B and Node O must decrypt the packet to view the intended recipient. When Node B sends the packet, Nodes A, C and O hear

the broadcast. All three nodes (A, C and O) must decrypt the packet intended for Node C. Thus to route one packet of data, the network as a whole must decrypt six packets when the minimum number of packets needed to be decrypted for routing is three. This results in half of the packets (or at the very least their associated headers with the routing information) being decrypted needlessly.

One possible way to solve this problem is to encrypt a unique destination id (IP address) and place it in the first bytes of a packet. A radio need only read those bytes, run a short decryption and check where said packet is destined; reference routing tables and see if it must take action on the packet. This has several disadvantages: All nodes in the network must share the same key. Key updating requires synchronization across an entire network. As ad hoc networking becomes more prevalent, nodes will be miles apart and will become separated; updating all nodes with the same key will present challenges. If a very large network, on the scale of a division, is using the same key to communicate on a logical network, this will generate a massive number of packets that can be intercepted and recorded. It is theoretically possible, given the small size and frequent reuse of the destination id, that the shared key, could be compromised, exposing the topology of the network.

Given knowledge of the exposed topology and the patterns of traffic, it becomes easier to classify the data types (UDP, HTTP, etc) by the latencies, sizes, messaging flow, node movements, etc. Then, given basic radiometric means (DF, signal strength, etc), the cracked node topology, communication types and flows between nodes; individual nodes and groups of nodes can be classified as to their type and function.

II. GOALS

The desire is to develop a means that will conform to the following rules:

1. The means must do no harm. It must expose no more data or information then previously exposed.
2. It must add minimal overhead in terms of data to the packet.
3. It must net less CPU cycles versus decrypting a packet header.
4. It should be designed to work with various different routing protocols (AODV and DSR at a minimum).
5. It should be easy to insert into existing routing stacks.
6. It must decrease the number of packets that the network as a whole must decrypt.
7. Should simplify the key management of a network.

III. MERI – MINIMALY EXPOSED ROUTING INFORMATION

Minimally Exposed Routing Information is a balance of exposing the minimum amount of routing information while still denying insight into a networks topology.

In the simplest form, a single bit is exposed, if the intended receiver has an even value for the last octet, then the bit is 0, if it is odd, the first bit is 1. As a result, the network's radios only need to examine the first bit of data, if the last octet of their IP address matches the even/odd value set in the packet, then they would receive, decrypt and examine the destination address to see if it was meant for them. Thus a network of at least size 4, given each radio is the destination for 25% of the packet traffic, a radio would only have to decrypt and examine 50% of the packets. This would decrease power consumption and increase the radio's and the network's performance. Exposing a second bit would reduce the theoretical number of packets needed for examination down to 25%.

It is assumed that for the network, given n bits exposed, there are $2^n * 2$ nodes in the network with equally distribution of exposed bit values and that over a series of packets, the exposed bits distribution is even. This describes an optimal network and data configuration defining the maximum theoretical savings, real world scenario savings would be less but still a significant savings.

The desired number of bits exposed is a factor of the number of nodes in the network verses the security consideration. In a four node (a very small number) network, as long as even distribution is assured, exposing a single bit does reveal some of the networks topology but might be an acceptable trade off verse performance. In the case of a very large network, more bits can be exposed without overly compromising security.

Utilizing a minimal number of exposed bits, an encrypted, wireless network's throughput can be increased and individual nodes can benefit from reduced processing and increased energy savings at a potential cost of security.

IV. ORI – OBSCURED ROUTING INFORMATION

The goal of ORI is to develop a means by which a radio can quickly identify packets destined for itself and only itself without having to decrypt any header or routing information yet still hide/protect said routing information.

Each time a broadcaster sends a packet, it embeds a pseudo random number/ODT (Obscured Destination Tag) of some size significant enough so that it is unlikely (although an ODT collision is acceptable) two neighboring broadcasters will generate the same value in time. This number is protected inside the encrypted packet; the broadcaster retains a list of pending ODTs. When a packet is received, it is decrypted and only the intended receiving node associates the embedded ODT with the IP of the broadcaster. Thus, next time the receiver must send a packet to the original broadcaster, the ODT/pseudo random number is placed, in clear text, at the head of the encrypted packet. Each node/radio in range reads the first X bits (size of the ODT) and references its own pending ODT list for a match. If it does match, the radio can then decrypt the packet and verify it is the intended recipient by the protected IP address. It then views the sender's new ODT and updates the lookup table. Every time a packet is received, a new ODT value is received in the decrypted packet and associated with the broadcaster. ODTs have a short life/time within the network and an eavesdropper, matching ever changing ODT's to nodes will be difficult.

In the case that two or more nodes happen to utilize the same ODT, each node will decrypt the packet and examine the internal IP, only the node whose IP address matches its own will then process the packet.

A special header value of all bits set can signify a complete "broadcast for all" in the case that a packet is meant for all or the broadcaster does not have an associated ODT for the intended receiver. Marking broadcast packets as such is a potential compromise of the network, but normally such messages are followed by a broadcast storm, a flurry of messages from neighboring nodes in response to the request. It is debatable if this does expose more information. Even if such is the case, ORI will still have a certain percentage of normal messages where the sending node will not have an ODT for the intended receiver. This will cause a number of packets being marked "broadcast for all," creating sufficient noise when trying to decipher between the two.

A. ORI Example

From the prior scenario, but this time leveraging ORI, Node A again wishes to send a packet to Node C with Node B acting an intermediary (A->B->C); in the neighborhood of Node B is Node O (observer). As the first packet sent, node A embeds ODT_{A1} inside the encrypted portion of the packet; it then retains this ODT_{A1} in a pending ODT table. Node A then sends the packet marked "broadcast for all" (all bits set) intended for Node B for routing to Node C; both Node B and Node O must decrypt the packet to view the intended recipient. When Node B decrypts the packet, it is able to collect the ODT_{A1} for Node A and stores this information. Node B will embed in the encrypted portion of the packet its own ODT_{B1} and then transmits the packet, marking it "broadcast for all." Nodes A, C and O will all observe the broadcast and all three nodes (A, C and O) must decrypt the packet marked "broadcast for all." When Node C decrypts the packet, it stores the ODT_{B1} sent to it from Node B. Node B now holds ODT_{A1} and Node C holds ODT_{B1}.

Now we extend the prior scenario; Node C sends a reply message to A with Node B acting as the router (C->B->A). Node C embeds its own ODT_{C1} into the packet prior to encryption, removes the held ODT_{B1} sent to it from the prior message (B->C) and places it, clear text, at the packet's head. Node C then broadcasts this packet and both nodes B and O will view the

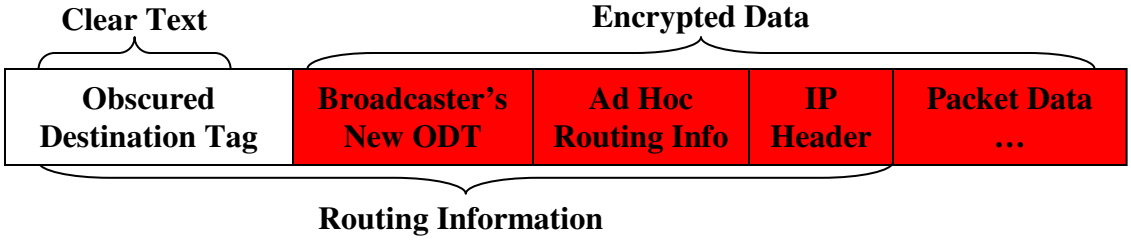

Figure 2: A basic diagram of an ORI data packet. When a sender must broadcast a packet, referencing an internal lookup table in memory, it finds the destination IP address and the pseudo random number associated with the intended receiver. This ODT (Obscured Destination Tag) is placed at the head of the encrypted packet and signifies the receiver within the network. Each time a packet is broadcasted, every receiver will examine the first N bytes equal to the length of the ODT; only the intended receiver will find a match for the ODT in a lookup table and will decrypt the packet. All other receivers will discard the packet as not being destined for them

transmission. Node O examines the first bytes containing the clear text ODT_{B1}, checks its pending ODT list, fails to find a match and rejects the packet. Node B likewise views the clear text ODT $_{B1}$, checks its pending ODT lists and finds a match. Node B then decrypts the packet, validates it is the intended recipient, removes ODT $_{B1}$ from the pending list and stores ODT_{C1}. B generates ODT_{B2}, places this value into the packet, encrypted the data, pulls ODT_{A1}, places this clear text at the packets head and broadcasts the packet. Nodes C and O both observe the clear text ODT_{A1}, reference their pending lists and reject the packet as not meant for them. Node A observes ODT_{A1}, finds the reference in the pending list, decrypts the packet and verifies it is the intended recipient.

In this scenario with four broadcasted messages (1. A->B, 2. B->C, 3. C->B, 4. B->A), without utilizing ORI, this round trip packet would have been decrypted needlessly six times. The first message by node O, the second by nodes A and O, the third by node O and the fourth message by nodes O and C. As seen in the second example, with ORI, one round trip message resulted in a 50% decryption cost savings for the network as a whole.

As seen in figure 2, continuous round trip communication between nodes A and C increase the packet decryption savings.

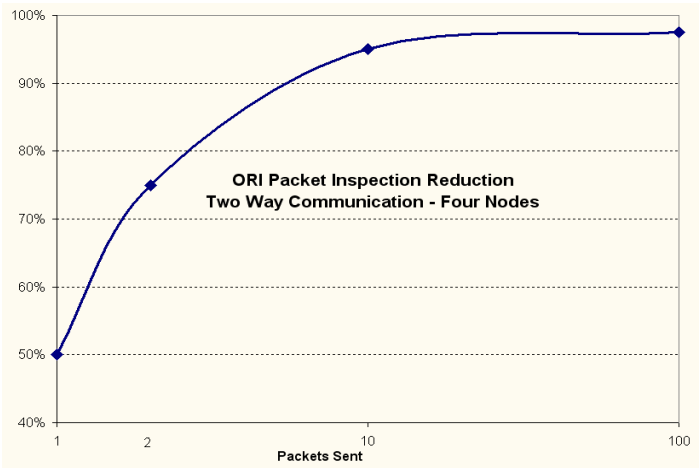

Figure 3: Scenario run using a NS2 ORI enhanced DSR protocol. This is a four node topology with A, B, C and O as described in the second scenario utilizing the topology of figure 1. Note: Only data packets in this simulation leveraged ORI and were used in the calculations, not route request type messages

B. ORI Caching

If a node receives multiple ODTs from a single source, it would be beneficial to retain these in a cache for future use; a cache of ODTs for any given node will increase effectiveness of ORI. The depth of such a caching strategy is a balance of implementation and storage complexity verses the increased network decryption reduction, most especially in small, embedded radios.

C. ORI Multi-Reuse

While potentially decreasing security, the ability of a single ODT to be used multiple times to address a single node would increase the effectiveness of ORI. The more times an ODT is reused increases a potential compromise of network security.

D. ORI Cache vs. Reuse

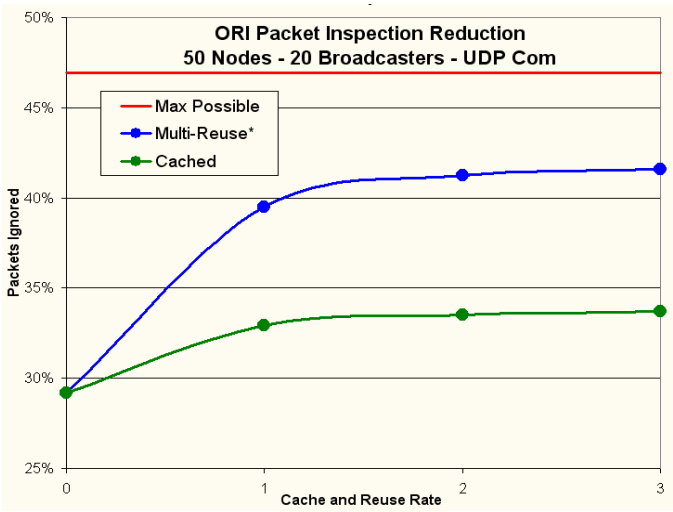

*Figure 4: ORI performance using a NS2 ORI enhanced DSR protocol with the included NS2 crb-50-20-4-512 and scen-670x670-50-600-20-0. The scenario has 50 nodes, with 20 of them broadcasting in four second intervals with random node movement. Note: Only data packets in this simulation leveraged ORI and were used in the calculations, not route request type messages. *Multiple reuse of ODT potentially compromises security.*

With NS2's crb-50-20-4-512 wireless DSR scenario, ORI managed an approximate 29.16% packet decryption reduction for the network as a whole as seen at cache depth and ODT reuse rates of 0. In this case, no ODT was used more then once and only a single ODT was kept for any given neighboring node. A FIFO cache was able to enhance effectiveness as the depth was increased from one (32.92%), two (33.51%) up to three (33.73%).

For this scenario, being able to reuse an ODT multiple times resulted in better performance over simply caching. At reuse rate of one (a single ODT used twice), ORI reduced the packet decryption count by 39.49%, 41.24% at reuse of two and three resulted in 41.58% of packets not needing to be decrypted. As a

baseline comparison, the maximum possible ORI success rate for this scenario's traffic was 46.96%. A combination of caching and reuse was not tested.

E. ORI Size

It is desirable to avoid ODT collision, where a single pending ODT could belong to two or more nodes. In this case, all nodes with said ODT will decrypt the routing information of the packet, only the node who the packet was truly destined for will then take further action. A node will have several pending ODTs, one from every neighboring node it has shared communication. If there are N nodes in a neighborhood, and fully meshed communication is assumed, then there would be $(N-1)^2$ pending ODTs in any given neighborhood. If the desired collision rate is ~1%, then I would require ODTs of length matching $(N-1)^2 * \sim 100$. Since two ODTs must be added to each packet, a clear text for the intended receiver and an encrypted ODT for the sender, the additional overhead at an approximate 1% ODT collision rate would be

$$(N-1)^2 * 100 * 2 \qquad (1)$$

F. ORI Implementation

ORI attempts to wrap existing packets, retaining existing protocol structures. This allows for ORI functionality to be inserted between the MAC and IP layers, just prior to encryption. It is independent of the protocol layers higher up the OSI stack.

V. COMPARISON

ORI has several advantages over MERI. First, it better protects the routing information since it is based on ever changing lookup table references. This presents a more difficult target analysis for an eavesdropper since the routing information is obscured. Second, since each ODT is basically unique to an intended target, network utilization is no longer limited by the slowest crypto subsystem in the network as only intended packets will need to be decrypted. This also has the effect on reducing power consumption and allow for more nodes to be added to the network without significantly increasing the incremental processing needed for each radio.

ORI's disadvantage to MERI is that it must keep a table in memory of known and pending Obscured Destination Tags and their corresponding IP addresses.

It is possible to use a combination of several methods to enhance effectiveness and minimize risks. A network shared key used to encrypt a node descriptor, as describe prior, offers the highest performance from a whole packet decryption count but has potential security and key management issues. The more often a key is used, the great risk that it can be compromised. A blend of the two means would reduce the times a key must be used; network key encrypted destination when no ODT is available and ORI when it is. Because an ODT is pseudo random number shared via a prior protected packet, an eavesdropper, trying to collect packets in an attempt to break the key, would be unable to differentiate between those packets using ORI and those using an encrypted destination.

It can be assumed the memory requirements and additional complexity of the ORI algorithm would result in less power and processing requirements over a fully encrypted packet network. ORI allows for efficient routing while still protecting network information, reducing the data that must be decrypted thus lowering power consumption and increasing the wireless network's effective bandwidth and the number of nodes it can support.

ACKNOWLEDGMENT

Thanks to Dr. David K. Murotake of SCA Technica and Dr. Luiz DaSilva of Virginia Tech for reviewing and commenting on the concepts of MERI and ORI.

Thanks to Bryan J. Hogan for his NS2 DSR FAQ and taking the time to answer some of my questions. His FAQ can be found at www.geocities.com/b_j_hogan/

REFERENCES

[1] J. Broch, D. A. Maltz, D. B. Johnson, YC. Hu, J. Jetcheva. "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols" in Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking. ACM Press, 1998 pp. 85-97

[2] S. R. Das, C. E. Perkins, M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks" 2000 www.cs.sunysb.edu/~mahesh/papers/pcm2001.pdf

[3] D. B. Johnson, D. A. Maltz "Dynamic Source Routing in Ad Hoc Wireless Networks" in Mobile Computing (ed. T. Imielinski and H. Korth), Kluwer Academic Publishers, Dordrecht, The Netherlands. http://citeseer.ist.psu.edu/johnson96dynamic.html

[4] P. Papadimitratos and Z.J. Haas. "Secure Routing for Mobile Ad Hoc Networks" SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 27-31, 2002

[5] C. E. Perkins, E. M. Belding-Royer, S. Das "Ad hoc On-demand Distance Vector (AODV) Routing" Experimental RFC 3561, July 2003, www.ietf.org/rfc/rfc3561.txt

[6] K. Sanzgiri, B. Dahill, B.N. Levine, E. Royer, and C. Shields. “A Secure Routing Protocol for Ad Hoc Networks” Technical Report 01-37, Department of Computer Science, University of Massachusetts, August 2001 http://citeseer.ist.psu.edu/ dahill01secure.html

[7] A. Schumacher, T. Luh “Comparison of the three protocols, AODV, DSR and TBRPF – Part 5 of 6” November 2004, www.cs.helsinki.fi/u/kraatika/ Courses/IPsem04s/slides/comp.pdf