

Secure Multi-Hop Infrastructure Access

Baruch Awerbuch¹ Reza Curtmola¹ David Holmer¹ Cristina Nita-Rotaru² Herbert Rubens¹

I. INTRODUCTION

Wireless networking today is predominantly used to provide mobile users with untethered access to fixed infrastructure. This allows users to move freely throughout the office or warehouse while remaining continuously connected with the office network and the Internet. While traditional access point devices currently provide this capability, they have a limited coverage range and thus many access points are required to provide coverage of a given area. One solution to this problem is to use a routing protocol that allows the users to traverse multiple hops to the nearest access point. This greatly expands the coverage range of each access point while simultaneously reducing costs and simplifying deployment.

Multi-hop infrastructure access has previously been proposed by [1] and [2]. However, the proposed solutions have only focused on routing without considering security. In this work we provide several extensions to the Pulse protocol [1] that defend against a large class of external attacks. The main contribution is providing strong adversarial resilience while maintaining low overhead.

The Pulse protocol was originally evaluated with an infrastructure only traffic pattern. Later work [3] evaluated the protocol under a more general mobile ad hoc network model with peer-to-peer traffic. The protocol was shown to achieve high delivery ratios under a wide range of network densities, mobilities, and traffic loads. The protocol operates with periodic flood initiated by the *pulse source* which creates a proactively updated spanning tree throughout the network. The periodic pulse flood exploits the communication concentration at the pulse source by providing every node in the network with a continuously updated route. The proactive pulse flood provides scalability to high levels of mobility. As the mobility level increases, many failures begin to occur throughout the network. In the Pulse protocol, all broken routes are repaired simultaneously within one pulse interval using one flood. In contrast, an on-demand protocol may initiate one flood for every broken active route, and a proactive link-state protocol may generate one flood per link failure. As the number of failures increases, this results in congestion due to the additional routing overhead, limiting the scalability of these protocols to high levels of mobility.

While the performance and scalability of the protocol has been validated in existing studies, no existing work to our

knowledge has addressed securing the protocol against adversarial attacks. In this work we present an efficient security framework for protecting the Pulse Protocol operation from external adversarial attacks.

II. PROBLEM DESCRIPTION

We consider an ad hoc network in which nodes obtain multi-hop access to fixed infrastructure (such as the wired network or Internet) by reaching the gateway (the pulse source).

We consider the following adversarial model: all authenticated nodes are trusted to perform the routing protocol correctly. The adversaries are unauthenticated nodes that attempt to disrupt the routing service by dropping, injecting or modifying packets. Specifically, our routing protocol has to withstand the following attacks: replay, black-holes, flood rushing, and wormholes. These are well-known attacks against wireless routing protocols.

We assume that each node in the network has a unique pre-established shared secret with the infrastructure access gateway (pulse source), which is used for authentication. The system must efficiently support adding or revoking nodes from the network.

III. PROTOCOL OVERVIEW

We present a modified version of the Pulse protocol, which ensures packets are securely routed under the described adversarial model. In order to provide a secure routing service, all the nodes in the network share a network wide symmetric key. This key is used to provide authenticity and integrity of the routing packets and is established and maintained using the key management scheme discussed below.

Our protocol establishes a reliability metric based on past history and uses it to select the best path. The metric is represented by link weights where high weights correspond to low reliability. Each node in the network monitors the performance of its links to its neighbors, and uses the resulting weights for selecting the shortest path. Faulty links are identified using secure acknowledgements on each link. Faulty links are avoided as their cost increases.

IV. PROTOCOL OPERATION

The protocol operates similarly to the original Pulse Protocol, however all packets include a nonce for reply protection, and are encrypted and authenticated using the network shared key. This prevents external adversaries from participating as nodes in the routing protocol. However, external adversaries can still create wormholes, flood rush small routing packets to increase the probability of wormhole formation, and then drop large data packets which results in a black-hole attack.

¹Department of Computer Science, Johns Hopkins University, 3400 N. Charles St. Baltimore, MD 21218 USA. 410-516-5298 {baruch, crix, dholmer, herb}@cs.jhu.edu

²Department of Computer Science, Purdue University, 250 N. University Street, West Lafayette, IN 47907. 765-496-6757 crisn@cs.purdue.edu

In order to protect against these attacks, a cryptographic acknowledgement is required for each data packet that traverses a link. This strategy is similar to the one used in ODSBR [4], however our adversarial model is different in that it excludes insider attacks. These acknowledgements allow the protocol to establish a secure loss-rate history for each link in the network. This secure loss-rate information is then used as a routing metric, similarly to ETX [5], which allows the protocol to select paths composed of reliable links.

V. KEY MANAGEMENT

In this section we briefly describe some issues related to the management of keys in the system. All the participating nodes share a network wide symmetric key, K_N , which is used to provide authenticity and integrity of the routing packets.

While the set of participating nodes is dynamic, the network wide key K_N should be shared only by non-revoked nodes. Whenever nodes are revoked, the pulse source needs to generate a new K_N and distribute it to the set of valid nodes, so that revoked nodes are no longer able to participate in the protocol.

Since each node in the network has a pre-established individual shared key with the pulse source, the trivial solution would be for the pulse source to unicast the new network key encrypted with the individual key to each node. However, this solution is inefficient. Instead, we use the LSD technique described in [6] for broadcast encryption. The pulse source plays the role of the center in the LSD scheme. Specifically, we define the *join* and *revoke* operations for nodes:

- *join* - to join the network, a node communicates with the pulse source using its individual pre-established shared key. Using this secure channel, the pulse source provides the node with both the current network key K_N , and a number of subset keys. The subset keys are used by the LSD scheme, and will allow the node to decrypt updated versions of K_N in the future.
- *revocation & key refresh* - to update the network key, the pulse source generates a new K_N , encrypts it using the LSD scheme, and broadcasts it to the network. The broadcast encryption semantics ensure that revoked nodes will not be able to decrypt the new network key.

In the LSD algorithm, the number of subset keys stored by each node is $O(\log^{1+\epsilon}(n))$, the length of the broadcast message is $O(r)$ (where r is the number of revoked nodes), while each node will have to perform $O(\log(n))$ cryptographic operations to decrypt the broadcast message (where n is the number of nodes in the network). The LSD scheme is practical, since it uses less than 1 kilobyte of storage for each node, and the pulse source can revoke any r out of 2^{28} nodes by sending on average a message of length $2r$.

VI. ATTACK ANALYSIS

Injecting: A network shared key prevents external adversaries from creating valid packets.

Modifying: An HMAC with the network shared key prevents an adversary from tampering with packets.

Replay: A nonce prevents an adversary from retransmitting previously transmitted packets on the network. The nonce is unique per node and replaced at every hop.

Wormhole: Wormhole creation is not prevented, however, the reliability metric allows nodes to avoid using wormhole links that do not deliver data packets.

Flood Rushing: Our protocol always prefers paths of lower metric regardless of order of arrival. Therefore flood rushing can be used to increase the probability of the routing protocol selecting a wormhole controlled link only when the metrics are equal.

Black-hole: Black-hole attacks are prevented by avoiding unreliable links.

VII. CONCLUSION

We presented a secure version of the Pulse Protocol for multi-hop infrastructure access. The protocol prevents a wide variety of external adversarial attacks. The protocol is light weight and relies solely on symmetric cryptography.

REFERENCES

- [1] B. Awerbuch, D. Holmer, and H. Rubens, "The pulse protocol: Energy efficient infrastructure access," in *Proceedings of The 23rd Conference of the IEEE Communications Society INFOCOM*, 2004.
- [2] S. Lee, S. Banerjee, and B. Bhattacharjee, "The case for a multi-hop wireless local area network," in *Proceedings of The 23rd Conference of the IEEE Communications Society INFOCOM*, March 2004.
- [3] B. Awerbuch, D. Holmer, and H. Rubens, "The pulse protocol: Mobile ad hoc network performance evaluation," in *Wireless On-demand Network Systems and Services 2005*.
- [4] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *ACM Workshop on Wireless Security (WiSe) 2002*, 2002.
- [5] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless networks," in *9th annual international conference on mobile computing and networking (MobiCom 03)*, September 2003.
- [6] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," in *Advances of Cryptology - Crypto '02*, vol. 2442 of LNCS, pp. 47–60, Springer-Verlag, 2002.