Ad Hoc Networks 79 (2018) 20–42

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Enabling individually entrusted routing security for open and decentralized community networks

Axel Neumann, Leandro Navarro*, Llorenç Cerdà-Alabern

Universitat Politécnica de Catalunya, Spain

ARTICLE INFO

Article history:
Received 30 October 2017
Revised 4 May 2018
Accepted 18 June 2018
Available online 19 June 2018

Keywords:
Routing
Trust
Decentralized security
Multi-topology
Cooperation
Mesh networks
Community networks

ABSTRACT

Routing in open and decentralized networks relies on cooperation. However, the participation of unknown nodes and node administrators pursuing heterogeneous trust and security goals is a challenge. Community-mesh networks are good examples of such environments due to their open structure, decentralized management, and ownership. As a result, existing community networks are vulnerable to various attacks and are seriously challenged by the obligation to find consensus on the trustability of participants within an increasing user size and diversity. We propose a practical and novel solution enabling a secured but decentralized trust management. This work presents the design and analysis of securely-entrusted multi-topology routing (SEMTOR), a set of routing-protocol mechanisms that enable the cryptographically secured negotiation and establishment of concurrent and individually trusted routing topologies for infrastructure-less networks without relying on any central management. The proposed mechanisms have been implemented, tested, and evaluated for their correctness and performance to exclude non-trusted nodes from the network. Respective safety and liveness properties that are guaranteed by our protocol have been identified and proven with formal reasoning. Benchmarking results, based on our implementation as part of the BMX7 routing protocol and tested on real and minimal (OpenWRT, 10 Euro) routers, qualify the behaviour, performance, and scalability of our approach, supporting networks with hundreds of nodes despite the use of strong asymmetric cryptography.

1. Introduction

Community mesh networks [1–3] are ideally open and decentralized structures, growing and evolving organically as network infrastructure is contributed, deployed, and configured by their participants. Typically, a deployment of such networks, such as Guifi.net [4] with more than 34,000 total active nodes, is structured in different network clouds [5], each consisting of up to hundreds of nodes, constituting autonomous systems (AS), operating its cloud-specific internal routing protocol (e.g. OSPF and, OLSR), and peering with neighbouring clouds via an exterior gateway protocol (e.g. BGP).

The operation of such networks is based on the principle of cooperation among the members. These communities usually have participation rules, such as a membership licence or peering agreement [6–8], that define their freedom, openness and neutrality. Nonetheless, current designs and implementations of mesh networks impose comprehensive technical definitions and restrictions to achieve functional data transit and end-to-end delivery among any pair of network nodes [2]. That includes the use of a specific routing protocol and routing metric so that nodes can consistently learn and inform about the state of the network and update their own routing tables. In practice, due to the lack of mature implementations, only a very few of the proposed routing protocols are used in real deployments or have been experimentally analysed [3,9–11]. Among them there are the AODV [12,13], Babel [14,15], BMX6 [16], the widely used OLSR [17–19], and batman-adv [20] protocol implementations.

One shortcoming of the current solutions is given by the lack of routing-security support that comes without introducing centralized dependencies (e.g. certificate authorities) [21], which would contradict with the open and the decentralized objectives of such networks. Another problem lies in the protocol requirements for unified parametrisation of metrics and policies to determine Quality of Service (QoS), routing, trust and security decisions for all network nodes [22]. Such a strong level of unification prohibits the usage of individually defined policies and limits its openness. It also imposes a substantial effort, increasing with the number and diversity of community members, for finding consensus on related questions.

* Corresponding author.
E-mail addresses: axel@ac.upc.edu (A. Neumann), leandro@ac.upc.edu (L. Navarro), llorenc@ac.upc.edu (L. Cerdà-Alabern).

https://doi.org/10.1016/j.adhoc.2018.06.014
1570-8705/

To dilute the limitations of a single and unified set of QoS parameters for routing, QoS multi-topology (MT) routing has been proposed [23,24], allowing the concurrent support of multiple virtual topologies (on top of a single physical topology), each established based on a different definition of QoS parameters. In OLSRv2 [19] MT is used for routers to support more than one link metric type. In this paper we use the MT concept to concurrently support different security and trust sets. A network could for example maintain one topology (a) for nodes trusted by organization A, and a second topology (b) usable only by nodes certified via organization B.

The security design of the protocol proposed in this work ensures that each node is the only authority able to define and publish its set of individually trusted nodes based on which forwarding rules (routes) for delivering its traffic should be selected, propagated, and maintained. This way, our protocol pursues the MT approach as it establishes dedicated virtual topologies for each participating node. It further supports the cooperative, open, and decentralized philosophy that enables community networking, as deployed network infrastructures remain open for other nodes to join and be used while being independent from any central entity.

This paper extends [25] the initial design and analysis of the securely-entrusted multi-topology routing (SEMTOR) protocol. In summary, the main contributions of this work are:

- Propose novel, secured, and decentralized routing protocol mechanisms called SEMTOR. With SEMTOR users can define individual trust sets of nodes, which are the only ones allowed to route their traffic.
- Summarize the assumptions, and respectively achieved safety and liveness properties and prove their correctness with formal reasoning.
- Describe how SEMTOR is implemented in BMX7 by extending BMX6 (BMX6 + SEMTOR = BMX7), a routing protocol currently used in production community wireless mesh networks [5].
- Experimental validation of the resistance of the SEMTOR implementation to various attack vectors and challenging network scenarios.
- Analysis of the performance and resource requirements of SEMTOR by measuring traffic, CPU, and memory overhead. The results demonstrate the scalability of SEMTOR to support existing mesh-network deployments and using inexpensive off-the-shelf WiFi hardware.

The remainder of this paper is organized as follows. After reviewing related work in Section 2, we identify the addressed problems and design objectives in Section 3 and detail the system model with further assumptions and definitions in Section 4. Section 5 describes the design and mechanisms of our protocol to solve these objectives which is then validated from an formal and an experimental perspective in Sections 6.3 and 7. This includes the presentation of our prototypical implementation and its functional and performance evaluation using embedded router hardware in a virtualized network environment. We discuss the contributions, open issues, and adjacent security solution in Section 8 and conclude in Section 9.

2. Background and related work

Existing work on secure routing for ad hoc and mesh networks has been reviewed in [26,27]. Authenticated routing for ad hoc networks (ARAN) [28] as proposed by Sanzgiri et al. and admittance-control enabling extensions for OLSRv2 proposed by Herberg et al. [29] use digital signatures to verify the authenticity and integrity of control messages. Both rely on the existence of a central certificate server trusted by all participating nodes. Babel hash-based message authentication code (HMAC) cryptographic authentication [30] relies on one or several pre-deployed shared keys to validate messages via attached message authentication codes (MAC). However, the requirement for preserving shared keys as a private secret within an open network community disqualifies related approaches for any open Community Network (CN).

In addition, SEAD [31] and SAODV [32] encounter the dependency on a central trust authority with a self-securing control plane. Using Anchored Hash Chain (AHCs) to protect the mutable hop counter field of routing- update messages they ensure that a malicious node cannot claim better distances to any remote node than it really has. However, both remain vulnerable to data-plane attacks such as packet dropping or routing-table poisoning. Moreover, SAODV proposes the use of digital signatures to protect non-mutable data in routing messages. To avoid the dependency of a certification authority as a central root of trust that guarantees the binding between node public keys (nodePKs) and other node properties such as their IP address Zapata [33] proposed to bind the identity of nodes given by their public key to their allocated address by building it based on the hash of the public key.

Work in [34,35] addressed the problem of misbehaving nodes by punishing malicious nodes based on their forwarding behaviour as observed and assessed by neighbouring nodes. Adnane et al. [34] build on top of SOLSR and extend it with detection and reaction mechanisms. Mogre et al. [35] present another holistic approach combining self-securing routing, detection, reputation, and counter-measure mechanisms.

Introduced in [25], SEMTOR follows a different approach. In fact, guaranteeing in all aspects the correct operation of nodes is indeed hard and, as pointed out by Adnane et al. [34], cannot be guaranteed (e.g. data-plane attacks cannot be prevented) by securing the topological information exchanged between nodes. Therefore, instead of aiming to ensure or enforce correct operation, SEMTOR enables each node admin to freely define their individual subset (and resulting sub-topology) from the complete set of participating nodes that the admin considers sufficiently trustworthy to meet the security and data-delivery objectives and concerns. In addition, none of the presented work relying on asymmetric cryptography for verification of control messages has yet been analysed in terms of performance or benchmarked based on embedded hardware and exposed to traffic and network characteristics that are typical for existing community mesh-network clouds. An overview of related work on routing security for IP-based mesh networks is given in Table 1.

An impressive amount of further related research about wireless mesh networks has been done in recent years. Selected publications are ordered thematically with respect to the importance for the objectives of this work. The case of community mesh networks is discussed in terms of legal implications, motivation, design, and business models in [1,7,21,39–42]. In addition, scalability and performance aspects of routing protocols are handled in [9,43–48]. Trust and security related work is surveyed and discussed in [27,49–52], with solutions for particular routing functions in [31,34,37,53–57], and presentations of holistic security frameworks in [35–37,58,59]. The last four also present measurement results based on simulation. Approaches towards supporting different or user-defined routing policies are handled in [22,60]. Traffic validation, or how to recognize a misbehaving path, node or link and which information is needed, is addressed by sketches [61–64], counters [65], fingerprinting [66] or sampling [62]. Distributed detection, the assessment of anomalous and faulty nodes based on sharing of distributed observations, considering the arbitrary behaviour of malicious nodes, is addressed by Π_2 and Π_{k+2} [67] in general, or by KDet [68], which is specifically for CNs (Table 2).