M. Sbeiti
C. Wietfeld
Communication Networks Institute,
Dortmund University of Technology

November 8, 2012

PASER: Position Aware Secure and Efficient Mesh Routing Protocol
draft-sbeiti-karp-paser-00

Status of this Memo

Copyright Notice

Abstract

The Position Aware Secure and Efficient Mesh Routing Protocol (PASER) aims to efficiently establish accurate routes in terms of metric and legitimated mesh nodes in wireless mesh networks in presence of external attackers. For this end, it achieves the following goals: Node authentication, message freshness and integrity, and neighbor transmissions authentication. The novelty of PASER lies essentially in combining asymmetric cryptography with Merkle tree (a lightweight cryptographic primitive) and a keyed-hash function to secure the routing messages. Another key feature of PASER is integrating (virtual) geographical positions of nodes in its hierarchical reactive routing process to enable an advanced network management while mitigating the wormhole attack. Apart from that, to address the problem of node compromise, PASER endorses a key revocation scheme to efficiently exclude those nodes.

Table of Contents

1. Introduction

Wireless mesh networks (WMNs) have recently become a promising technology to establish a high-performance and low-cost network anywhere anytime without the need for an existing infrastructure. To establish WMNs, routing protocols are necessary to discover and maintain routes on the fly between all network nodes. The latter makes WMNs prone to a new type of attacks [4], e.g., the wormhole attack. For instance, a pair of malicious nodes linked via a fast transmission path (e.g., Ethernet) forward route discovery messages faster than legitimated nodes. This causes victim nodes to always use the tunneled route to transmit their data packets, which are then dropped by the attacker. Even if the network is protected via conventional cryptosystems e.g., IEE802.11i in pre-shared key mode [5], this attack still succeeds. The main reason for this is that routing messages are simply forwarded, without any changes, from one end to the other end of the tunnel.
Thus, without a satisfactory level of security, end-users or organizations lack motivation to utilize this communication system. Otherwise, malicious users, terrorists or benefiting organizations might easily disrupt the communication channel. To address this issue, many approaches to secure routing in WMNs have been recently proposed [6]. However, none of these protocols has been adopted in the practice. The high overhead of the security mechanisms of these protocols or the hard assumptions taken by their design burdened their deployment in real life applications. For this end, PASER [2]

has been designed, to achieve a reasonable trade-off between security and performance as demonstrated in [3].

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, parameters enclosed by "<>" should be replaced with the appropriate value. The "/" symbol denotes a disjunction. The "^" symbol indicates the power function

This document defines the following terminology:

2.1. Terminology

Reactive
: A protocol operation is considered "reactive" if it is performed on-demand, in reaction to specific events. This terminology is adopted from [7].

Hierarchical
: Protocol architecture is called "hierarchical" if nodes have different role and thereby different behavior.

Node
: It is either a mesh router or a mesh access point or a mesh gateway.

Mesh Router (MR)
: MR is an entity that runs a routing protocol in order to offer routing services for other nodes / stations.

Mesh Access Point (MAP)
: MAP is an entity that has mesh router functionality and provides network access to associated stations.

Mesh Gateway (MG)

MG is an entity that has mesh router functionality and provides access to the Internet. Besides, a mesh gateway MUST have a secure connection to the key distribution center.

Station
A station is an entity that is a singly addressable instance of a medium access control (MAC) and physical layer (PHY) interface to the wireless medium. This terminology is adopted from [8].

Originator / Originating Node
The originating node is the data source node; it is the node that initiates a PASER route discovery process, i.e., it the node that creates a UB-RREQ message. This terminology is adopted from [7, 9].

Destination / Destination Node
It is the final target of a message. It is the node to which data packets are to be transmitted. This terminology is adopted from [9].

Forwarder / Forwarding Node / Intermediate Node
A forwarder is a node that should forward packets / messages destined for another node, by retransmitting / rebroadcasting them. This terminology is adopted from [9].

Sending Node / Sender
It is either an originator or a destination node or a forwarder. It is the node that sends the message.

Next Hop / Neighbor Node
A node X is a neighbor node of node Y if node X is in the transmission range of Y and the latter can hear node X, i.e., X is one hop far from Y. This terminology is adopted from [10].

Broadcasting
Broadcasting means transmitting to the network broadcast address. A broadcast message may not be blindly forwarded, but broadcasting is useful to enable dissemination of PASER messages throughout the ad-hoc network. This terminology is adopted from [9].

Flooding

In this document, flooding a message refers to the process of blindly broadcasting the message to every PASER node in the network. This terminology is adopted from [7].

PASER Interface
It is an interface the PASER protocol uses to exchange messages with other nodes.

Sub-network / Sub-network address
A PASER node may comprise several interfaces configured with different IP addresses. For instance, in case of a mesh access point, stations which require the services of the mesh access point are typically attached to another interface than the PASER interfaces and assigned IP addresses according to the class less inter-domain routing. Sub-network address is the network address of the IP address of all node interfaces but the PASER interfaces.

Distance
It is an unsigned integer which measures the distance between two nodes in meters.

Sequence Number
A Sequence Number is an unsigned integer (a monotonically increasing number) maintained by each PASER node. This sequence number guarantees the temporal order of routing information to avoid route-loops. The value zero indicates that the sequence number for a destination address is unknown. This terminology is adopted from [7].

Invalid route
An invalid route is a route that has expired, denoted by a state of invalid in the routing table entry. An invalid route is used to store previously valid route information for an extended period of time. An invalid route cannot be used to forward data packets, but it can provide information useful for route repairs, and also for future route request messages. This terminology is adopted from [9].

Valid / active route
A valid route is a route towards a destination that has a routing table entry marked as valid. Only valid or active routes can be used to forward data packets. This terminology is adopted from [9].

Key Distribution Center (KDC)

It is a logical unit responsible for the key management in PASER.

Group Transient Key (GTK)
GTK is a temporal key that is used among a group of nodes as basis for identifying a group member.

Client Transient Key (CTK)
CTK is a temporal key that is used between mesh access points and stations as basis for identifying one another.

Packet
It is a formatted unit of data exchanged between applications.

Message
It is a formatted unit of information exchanged between routing protocols such as PASER.

Trusted Neighbor
It is a neighbor that finished a trust establishment three-way handshake.

Trusted Broadcast (TB) / Trusted Unicast (TU)-<Message>
These are messages exchanged between trusted neighbors. They are secured with the PASER symmetric authentication scheme and the keyed-hash function.

Untrusted Broadcast (UB) / Untrusted Unicast (UU)-<Message>
These are messages exchanged between new neighbors. They are secured using digital signature.

External Attacker
It is an attacker that does not possess a valid PASER certificate.

2.2. Abbreviations

CRL - Certificate Revocation List

CTK - Client Transient Key

GTK - Group Transient Key

KDC - Key Distribution Center

MAP - Mesh Access Point

MG - Mesh Gateway

MR - Mesh Router

TB-Hello - Trusted Broadcast Hello

TB-RERR - Trusted Broadcast Route Error

TU-RREQ - Trusted Unicast Route Request

TU-RREP - Trusted Unicast Route Reply

TU-RREP-ACK - Trusted Unicast Route Reply Acknowledge

UB-Key-Refresh - Untrusted Broadcast Key Refresh

UB-Root-Refresh - Untrusted Broadcast Root Refresh

UB-RREQ - Untrusted Broadcast Route Request

UU-RREP - Untrusted Unicast Root Reply

WMNs - Wireless Mesh Networks

3. Applicability Statement

PASER is a suitable solution for wireless mesh networks (WMNs) with specific security requirements. It is mainly tailored for rescue organizations in emergency operations. In such environments, public (cellular) networks are typically either destroyed or overloaded and dedicated emergency services / networks such as TETRA suffer from insufficient data rates. WMNs, however, provide robust and reliable self-organizing, self-configuring and self-healing wireless broadband service access.
Nonetheless, PASER is not restricted to emergency scenarios; it is generally applicable as it does not make restrictive assumptions on the network nodes. Besides, it provides generic metrics for the constituent links of the discovered routes, allowing the implementation of any route selection algorithm.

PASER handles a wide variety of mobility patterns by dynamically determining routes on-demand. PASER also handles a wide variety of traffic patterns with the focus lying on traffic destined to the mesh gateway, i.e., to the Internet.

PASER supports nodes with multiple interfaces. In addition to routing for their local processes, PASER nodes can also route on behalf of other stations reachable via those interfaces. Any such station MUST NOT be served by more than one PASER node.

PASER only utilizes bidirectional links.

The routing algorithm in PASER operates at the network layer. Nonetheless, it may be operated at layers other than the network layer, using layer-appropriate addresses.

PASER REQUIRES a key distribution center that MUST be installed in a secure place and MUST have secure connection to mesh gateways.

PASER REQUIRES that legitimated nodes stick to the protocol behavior. It combats external nodes from attacking the network especially the routing functionality of WMNs. It is robust against impersonation attack, man-in-the-middle attack, replay attack, tempering attack and thus to the blackhole attack. The definition of these attacks is provided in [6].

To mitigate wormhole attack, PASER uses geographical leashes as proposed in [11]. Nevertheless, PASER nodes must not be placed outdoor in order to be aware of their geographical positions. They might be placed indoor und assigned virtual geographical positions by using the method described in [12].

PASER has been designed to achieve a reasonable trade-off between security and performance in order to gain acceptance in the practice and in order to develop a scalable secure routing protocol.

4. Protocol Overview

Using a hierarchical reactive routing approach and a concise combination of security mechanisms, PASER aims to efficiently establish accurate routes in terms of metric and legitimated mesh nodes in WMNs in presence of external attackers. An overview of the PASER operations is given in the next subsections.

4.1. Routing

4.1.1. Route Discovery

PASER is a hierarchical reactive routing protocol, which differs between mesh gateways, mesh routers and mesh access points. Before joining the network, all nodes are responsible to register themselves at the key distribution center (KDC). For this end, mesh routers / access points must always discover a route to a mesh

gateway in order to contact the KDC. Figure 1 below illustrates this process. It gives an overview of how a new mesh router / access point S, wanting to register itself at a KDC, performs the route discovery to a mesh gateway G to request, among others, the network keys from the KDC. Hereby, in addition to the information a mesh router receives from the KDC, a mesh access point receives the client transient key (CTK).

```
Routing Table of S (After Registration)
Destination: G X W Y Z
Next Hop   : W W W Z Z


            +-----------+                TU-RREQ           TU-RREQ
+-----------+UB-RREQ    +----->          +---->        +---------+
|   +-------+UU-RREP    +-----+ MR/MAP W <----+ MR/MAP X <-----+ |
|   |   +---+TU-RREP-ACK+---------^                            | |
|1  |2  |3  +-----------+                                     2|1|
|   |   |                                                      | |
+   v   +                                                      + v
New MR/MAP S                    Key Distribution+--+Mesh Gateway G
+   ^   +                            Center                    ^ +
|   |   |   +-----------+                                      | |
|   |   +---+UB-RREQ    +----->          +---->        +-------+ |
|   +-------+UU-RREP    +-----+ MR/MAP Z <----+ MR/MAP Y <-------+
+-----------+TU-RREP-ACK+--------^       TU-RREP           TU-RREP
            +-----------+
            Trust Establishment
            Three-way Handshake
```

Figure 1: Route discovery of PASER during the registration of a new mesh router / access point.

As this figure shows, PASER adopts the path accumulation approach (forwarding nodes append their own address to each route discovery message). Furthermore, destination nodes reply to all received requests. The figure also depicts that in PASER, new neighbors establish a trust relationship between each other. Afterwards, they mainly communicate via unicast messages.

4.1.2. Route Maintenance

Apart from specific timeouts defined for an existing route (see section 7), to detect and react on broken links, a node deletes a broken route in two cases. First, if it has not received a predefined number (e.g., 2) of trusted broadcast Hello messages from a neighbor. Hello messages are periodically exchanged between neighbors. Second, it did not get a link layer acknowledge for a

unicast packet sent to that neighbor, even after several retransmissions, e.g. 7 times, which is the default number of a frame retransmission according to IEEE802.11 [8]. While the link layer feedback enables the fast reaction of PASER on route breaks in case of active data transfer, Hello messages allow the detection of route changes also in case of no data transfer. Besides, Hello messages enable a proactive detection of nodes that are tow-hop far since they incorporate a neighbor list. In addition, Hello messages endorse the geographical position of the sending node, enabling a permanent update of neighbor's position, which is relevant for advanced network management, and which is necessary for protection against wormhole attack.

Upon receiving a packet for a destination the entry of which has been deleted or the next hop on the route to that destination is not available anymore, a forwarding node broadcasts a route error (TB-RERR) message in the network. This message comprises the last known sequence number and the IP-address of the unreachable next hop as well as all the nodes for which the unreachable node was the next hop, if available. When a node receives a TB-RERR message, it checks whether the sequence number of the unreachable node is fresh, and if the sender of the TB-RERR is the next hop to the unreachable node. Only if both requirements are met, the route is marked as invalid and the node rebroadcasts the route error message. This TB-RERR propagation mechanism enables more efficient network topology awareness in comparison to a simple flooding.

4.2. Security

PASER combines digital signature with lightweight authentication tree and keyed-hash function to secure the routing messages. Besides, to address the problem of node compromise, PASER endorses a key revocation scheme to exclude those nodes in a fast and efficient way. For this purpose, it supports a dynamic distribution of network keys. The main building blocks of PASER's security are depicted in the Figure 1 below. They are applied as follows.

```
                +-------------+ PASER +--------------+
                |                                    |
                |          Security Systems          |
   +------------v---------------+   +----------------v---------------+
   |Asymmetric-Key Cryptography |   |Symmetric-Key Cryptography      |
   |(Certificates with          |   |(Group/Client Transient Key)    |
   | Integrated Roles)          |   |                                |
   +------------+---------------+   +----------------+---------------+
                |         Security Mechanisms        |
   +------------v---------------+   +----------------v---------------+
   |Key Management    RSA       |   |Merkle Tree  Keyed-Hash Function|
   |Digital Signature           |   |                                |
   +------------+---------------+   +----------------+---------------+
                |         Security Primitives        |
   +------------v---------------+   +----------------v---------------+
   |Certificates with           |   |Tree Leafs (Secrets)            |
   |Integrated Roles            |   |Group/Client Transient Key      |
   +------------+---------------+   +----------------+---------------+
                |          Secure Messages           |
   +------------v---------------+   +----------------v---------------+
   |UB-RREQ    UB-Root-Refresh  |   |TU-RREQ     TU-RREP             |
   |UU-RREP    UB-Key-Refresh   |   |TU-RREP-ACK TB-RRER TB-HELLO    |
   +----------------------------+   +--------------------------------+
```

Figure 2: Security mechanisms endorsed in PASER.

4.2.1. Digital Signature Scheme

It is used for the authentication of broadcast-messages; to establish trust between new neighbors. Hereby, a node uses the key pair bound to its certificate. Certificates SHOULD have integrated roles e.g., mesh gateway, router or access point. These roles are for instance included in the extension area of an X.509 certificate. They reflect predefined responsibilities of a node in the network and, thus, they map the hierarchy of a mesh network. Apart from that, digital signature is used by the KDC to sign the information it sends to the nodes. Recommendations listed in [14] should be considered when selecting a digital signature scheme.

4.2.2. Symmetric Authentication Scheme

It is based on the Merkle tree [13] to authenticate unicast-messages between trusted neighbors. Each mesh node generates 2^n random secrets, where n is a configuration parameter that depends on the use case scenario. The generated secrets are the leaf pre-images of the authentication tree. These are used to compute the tree root element as described in [2, 3, 13]. After computing the root

element, a node broadcasts it to its neighbors. As a next step, any mesh node e.g., S, wanting to send a routing message to a neighbor W, discloses a secret (e.g., secret1) and sends it along with the corresponding tree path and the routing message to that neighbor W. Fourth, the neighbor W, already knowing the root element of the mesh node S, computes the root of the secret it has received and verifies, if it matches the root of the mesh node S. If true, the neighbor W can trust that the message has been sent by the mesh node S.

PASER tree's secrets are l bits long, where l is a configuration parameter and l > n. A secret SHALL be constructed as follows: The least significant (l - n) bits are generated randomly for each secret. The most significant n bits constitute an initialization vector (counter), the value of which is 0 for the first secret. The initialization vector is incremented by one for each subsequent secret.

Upon disclosing (2^n - 1) secrets during the network lifetime, a node must generate a new root element. The latter guarantees the freshness of a secret. That is, a secret value can never be used twice for a given root. This technique is used to prevent replay attacks.

4.2.3. Keyed-Hash Function

It is applied to guarantee the integrity of unicast-messages based on a group transient key. This function is always used in combination with the lightweight symmetric scheme to secure PASER messages between trusted neighbors. Recommendations listed in [14] should be considered when selecting a keyed-hash function.

4.2.4. Key Management Scheme and RSA

The dynamic distribution of the group transient key and the mesh access client first occurs at network setup, when a node registers itself for the first time at the KDC. The mesh gateway forwards a MR/MAP request or sends itself a request to a key distribution center (KDC) over a secure channel. The KDC responds to that request by sending the network keys encrypted with the node's public key using the RSA algorithm. Hereby, a nonce is used to guarantee the freshness of the messages. Besides, a Key-To-Use mark is also sent to that node. Key-To-Use mark is the number of the key in use signed by the KDC. Nodes always include the number of the key in use in each PASER unicast message. This number is increased by one for each new generated key. The key is regenerated in case a node gets compromised. In that case, a new Key-To-Use mark, initialized by the KDC, is flooded in the network, and the certificate of the

compromised node is blacklisted. Upon receiving the new mark, each node resets its PASER tables and re-registers itself at the KDC. If a legitimated node was meanwhile unreachable, the node detects from the higher key number in use that key refreshment has occurred. Neighbors of that node even prove the latter using the new Key-To-Use mark. As a result, that node goes in a reset state. Due to the Key-To-Use mark, an attacker, who compromised a node, cannot denial the service of neighbor nodes by just increasing the key number of its message.

Recommendations listed in [14] should be considered when selecting the RSA parameters. Note that any other algorithm than RSA could be used to encrypt the keys sent from KDC to a node if it fulfills the confidentiality goal.

5. Messages Format

PASER comprises four untrusted messages: UB-RREQ, UU-RREP, UB-Root-Refresh and UB-Key-Refresh and five trusted messages TU-RREQ, TU-RREP, TU-RREP-ACK, TB-RERR and TB-Hello. The format of these messages is illustrated in Table 1 and Table 2, respectively. These messages are composed of some of the following fields:

Basic Fields

- o Type
  1. UB-RREQ
  2. UU-RREP
  3. TU-RREP-ACK
  4. TU-RREQ
  5. TU-RREP
  6. TB-Hello
  7. TB-RERR
  8. UB-Root-Refresh
  9. UB-Key-Refresh

- o Timestamp
  It reflects the creating time of the message. It is used to combat replay attacks.

- o Registration Flag (R)
  It is set if a node wants to register itself at the key distribution center in order to join the network.

- o Mesh Gateway Flag (G)
  It is set if the route discovery destination is a mesh gateway.

- o Originator IP Address
  It is the IP address of the node that started the route discovery.

- o Destination IP Address
  It is the IP address of the route discovery destination.

- o Originator Sequence Number
  It is the sequence number of the node that initiated the route discovery.

- o Destination Sequence Number
  Sequence number of the route discovery destination.

- o Forwarder Sequence Number
  It is the sequence number of the node that forwarded the message.

- o Metric Originator <-> Sender
  It is the metric between the node that started the route discovery and a sender node.

- o Metric Destination <-> Sender
  It is the metric between the route discovery destination and a sender.

- o Route Address Range List
  List of IP addresses of the PASER interfaces and all sub-networks of interfaces other than the PASER interfaces which a node comprises. Each node that forwards a message appends this information to the list.

- o Neighbors Address Range List
  List of neighbors IP addresses and the sub-networks for which neighbors are responsible.

- o Unreachable Destinations IP Addresses List
  It is a list of IP addresses and sequence numbers of nodes that are not reachable anymore.

Neighbor Identification Fields

- o Originator Nonce
  Random number created and sent during the registration of a node. This nonce protects against man-in-the-middle or replay attacks during the registration.

- o Originator Certificate
  It is the certificate of the node that started the route discovery.

- o Destination / Forwarder Certificate
  It is the certificate of the node that forwarded a message or of the destination when sending a reply to neighbors.

- o Sender Root
  Root element of the sender node.

- o Sender Initialization Vector
  It is the current value of the initialization vector of the sender node.

- o Originator Position
  It is the current position of the node that started the route discovery.

- o Forwarder Position
  It is the current position of the node that forwarded the message.

- o Destination Position
  It is the current position of the destination of the route discovery.

- o Group Transient Key Number
  Current number of the group transient key in use.

- o Key Distribution Center (KDC) Block
  - o Encrypted Group Transient Key
    Group transient key encrypted with the public key of the originator.

- o Encrypted Client Transient Key
  Client transient key encrypted with the public key of the originator.

- o Originator Nonce
  Random number created and sent by the originator during the registration process.

- o Certificate Revocation List
  It is a list of all revoked certificates.

- o Group Transient Key Number
  Number of the group key currently in use.

- o KDC Certificate
  It is the certificate of the KDC.

- o KDC Signature
  Signature (using the KDC private key) of all elements of the KDC block.

- o Key Distribution Center Certificate
  It is the certificate of the key distribution center.

- o Key Refresh Signature
  Signature (using the KDC private key) of all the elements of an UB_Key_Refresh message.

- o Sender Signature
  It is the signature using the private key of the node that sent the message. The sender signs all elements of a message.

Neighbor Authentication Fields

- o Sender Secret
  It is the current secret of the node that sent the message.

- o Secret Authentication Path
  It is the authentication path of the enclosed secret.

- o Keyed-Hash

It is a the keyed-hash value of all elements of a message. The hash value is calculated using the group transient key (GTK).

Notations in Table 1 and Table 2:

x: indicates that a message comprises a field.

*: indicates that a message comprises a field if the Registration flag is set.

var.: means that a field has a variable length. Fields having variable lengths are typically preceded by 4 Bytes in which the current length of the field is stated.

c.: is an estimation of the length of a field. The estimations apply in case RSA with modulo size of 1024 bits is used for encryption and signature, and SHA 256 is used for hashing. These estimations comprise the 4 Bytes preamble for each field having a variable length.

m(<value>): means that the length is a multiple of value.

<value>.#<parameter>: means that the length equals value times the number of parameter

<flag2>|<flag1>: means flag1 is the least significant bit and flag2 is the next bit.

Table 1: Format of Untrusted Messages

Field	Size [Byte]	UB-RREQ	UU-RREP	UB-Root Refresh	UB-Key Refresh
Basic Fields					
Type	1	x	x	x	x
Timestamp	4	x	x	x	x
Registra./ MG Flags	1 F|R	x	x		
Originator IP Address	16	x		x	x
Destination IP Address	16	x	x		
Originator Sequence Number	4	x	x	x	
Destination Sequence Number	4		x		
Forwarder Sequence Number	4	x			
Metric Originator Sender	1	x	x		
Metric Destination Sender	1		x		
Address Range List	var. m(16)	x	x		
Neighbor Identification Fields					
Originator Nonce	4	*			

Originator Certificate	var. c.701	*			
Forwarder / Destination Certificate	var. c.701	x	x	x	
Sender Root	32	x	x	x	
Sender Initializa-tion Vector	var. c.4	x	x	x	
Originator Position	8	x		x	
Forwarder Position	8	x	x		
Destination Poistion	8		x		
Group Key Number	4	x	x		x
KDC Block	var. c.1604		*		
KDC Certificate	var. c.744				x
Key Refresh Signature	var. c.132				x
Sender Signature	var. c.132	x	x	x	

Table 2: Format of Trusted Messages

Field	Size[B]	TU-RREQ	TU-RREP	TU-RREP-ACK	TB-RERR	TB-Hello
Basic Fields						
Type	1	x	x	x	x	x
Registra./ MG Flags	1 F|R	x	x			
Originator IP Address	16	x	x	x	x	x
Destination IP Address	16	x	x	x		
Originator Sequence Number	4	x		x	x	x
Destination Sequence Number	4		x			
Forwarder Sequence Number	4	x				
Metric Originator Sender	1	x	x			
Metric Destination Sender	1		x			
Route Address Range List	var. m(16)	x	x			
Neighbors Address Range List	var.					x
Unreachable Destinations	20.# Unreac.				x	

IP Addresses List	Destin.					

Neighbor Identification Fields

Originator Nonce	4	*				
Originator Certificate	var. c.701	*				
Originator Position	8	x				x
Forwarder Position	8	x	x		x	
Destination Poistion	8		x			
Group Key Number	4	x	x	x		
KDC Block	var. c.1604		*			

Neighbor Authentication Fields

Sender Secret	32	x	x	x	x	x
Secret Authentica- tion Path	32.# Secrets	x	x	x	x	x
Keyed-Hash	32	x	x	x	x	x

6. Tables Structure

Nodes running PASER maintain two tables namely, a routing table and a neighbor table. These tables are defined as follows.

Routing Table

- o Destination IP Address
  It is the IP address of the route destination.

- o Neighbor IP Address
  It is the IP address of the next hop towards the route destination.

- o Route Delete Timer
  Route will be deleted when this timer expires.

- o Route Invalidate Timer
  Route will be invalidated when this time expires. An invalid route cannot be used but it can be restored faster than a route discovery.

- o Destination Sequence Number
  It is the current sequence number of the route destination.

- o Metric
  It is the metric between this node and the route destination.

- o Destination-is-Mesh-Gateway Flag
  Flag is set if the route destination is a mesh gateway.

- o Route-is-Valid Flag
  Flag is set if the route is still valid.

- o Destination Sub-Networks List
  All sub-networks of the route destination.

- o Destination Certificate
  It is the certificate of the route destination.

Neighbor Table

- o IP Address
  It is the IP address of the neighbor.

- o Delete Timer
  When this timer expires the neighbor will be deleted from this

table and all route entries for which this neighbor is next hop will be deleted from the routing table.

- o Invalidate Timer
  When this timer expires the neighbor is set as invalid and all route entries for which this neighbor is next hop will be set as invalid.

- o Trust Flag
  This Flag is typically set during / after the trust establishment three-way handshake between neighbors. It reflects the current trust relation between them.

- o Neighbor-is-Valid Flag
  This flag is set if the neighbor is considered valid.

- o Root
  Root element of the neighbor.

- o Initialization Vector
  It is the current value of the neighbor initialization vector.

- o Position
  It is the current position of the neighbor.

- o Certificate
  It is the certificate of the neighbor.

- o Interface Index
  It is the index of the interface over which the neighbor is reachable.

7. Timers

PASER comprises the following timers:

- o Route_Discovery_Timeout
  It is the maximum time an originator node waits for a route reply. When this timer expires, the route discovery will be restarted and the timer will be refreshed until a maximum number of repetitions are reached. In that case, saved packets for that destination are dropped.

- o Route_Entry_Delete_Timeout
  When this timeout is triggered, the corresponding route entry is deleted from the routing table. In case the route is valid, this timer is refreshed every time a node receives or sends a PASER message or a data packet over the route. In case the route is invalid, this timer is refreshed only if the node receives a PASER message over the route. In that case, the route is set as valid again.

- o Route_Entry_Invalidate_Timeout
  When this timeout is triggered, the corresponding route entry is set as invalid in the routing table. This timer is refreshed every time a node receives or sends a PASER message or a data packet over the route. An invalid route gets valid again in case a node receives a PASER message over the route before deleting it. In that case, this timer is reset.

- o Neighbor_Entry_Delete_Timeout
  When this timeout is triggered, the corresponding entry is deleted from the neighbor table. The corresponding entries in the routing table SHOULD be also deleted. This timer is refreshed upon receiving a PASER message form this neighbor.

- o Neighbor_Entry_Invalidate_Timeout
  When this timeout is triggered, the corresponding entry is set as invalid in the neighbor table. The corresponding entries in the routing table SHOULD be also set as invalid. This timer is refreshed upon receiving a PASER message form the neighbor. An invalid neighbor is set to valid again if trusted message is received from that neighbor. Else, a route discovery is required if data packets need to be send to this neighbor.

- o Root_Resend_Timeout
  It is the time a node waits before resending its new root element. A new root element is typically broadcasted three times.

- o Hello_Periodic_Broadcast_Timeout
  It corresponds to the Hello interval between two successive Hello messages. This timer is refreshed after sending a hello message.

- o KDC_Request_Timeout

It is the maximum time a mesh gateway node waits for a KDC reply. When the timer expires, the gateway resends the request. There is no upper limit for the number of retransmissions of this request.

- o TU_RREP_ACK_Timeout
  It is the maximum time a node waits for a TU-RREP-ACK in order to finish the trust establishment three-way handshake. When this timeout is triggered, a node resends the UU-RREP message. The maximum number of UU-RREP retransmissions SHOULD be set to three.

- o Key_Refresh_Timeout
  It is the maximum time a KDC waits before refreshing the network group transient key, i.e., before sending a UB-Key-Refresh message.

8. PASER Operations

8.1. Registration at the Key Distribution Center (KDC)

At power-up and before any communication can take place, a node undergoes the following steps in order to join the network:

1. It generates empty routing and neighbor tables according to section 6.

2. It sets its sequence number to 1.

3. It generates the Merkle tree secrets and computes the root element as described in [2, 3].

4. It executes the following depending on its type or the role it is assigned in its certificate:

    1. Mesh Gateway: It requests group and client transient keys as well as a certificate revocation list and the number of the current key in use from the key distribution center. Upon receiving the reply, the mesh gateway enters the registered state. To prevent replay attacks, the mesh gateway includes in the request a nonce, which gets signed by the KDC in the reply. We do neither restrict the choice of the protocol used to request this information nor the location of the KDC. We assume however that the communication between a mesh gateway and the KDC is secure. Apart from that, we assume that the KDC is placed in a safe location. Since the KDC is a logical unit, it

can be installed anywhere. For instance, in emergency and rescue operations, it is reasonable to install the key distribution center on the main mesh gateway, which is placed on top of the fire-fighting command and control vehicle.

2. Router/Access Point: It starts a route discovery towards a mesh gateway as part of the registration process. Hereby, the node also (like the mesh gateway) sends a nonce to prevent replay attacks. When the request arrives at a mesh gateway and the Registration flag is set, the mesh gateway forwards the registration request to the KDC and it replies the KDC reply to the node. Upon receiving the KDC block, a node enters the registered state. It possesses the keys required to join the network.

5. It sends a Hello message and initializes the Hello_Periodic_Broadcast_Timeout.

8.2. Tables Management

Upon receiving a PASER message that passed all verification checks (see section 8.5), a node undergoes the following steps with respect to PASER tables:

1. Neighbor Table

   - Receiving of Untrusted Broadcast Route Request (UB-RREQ): The node verifies if the sender of the message has an entry in the neighbor table. If not, create a new entry with the corresponding information and timers and set the Neighbor-is-Valid flag to 1 and unset the Trust flag to 0. If the sender already has an entry in the neighbor table, verify if the neighbor is valid. If it is not, set Neighbor-is-Valid flag to 1 and unset the Trust flag to 0. Refresh timers and the corresponding entry information.

   - Receiving of Untrusted Unicast Route Reply (UU-RREP): The node verifies if the sender of the message has an entry in the neighbor table. If not, create a new entry with the corresponding information and timers and set the Neighbor-is-Valid flag and the Trust flag to 1, respectively. If the sender already has an entry in the neighbor table, refresh timers and the corresponding entry information. Set the Neighbor-is-Valid flag and the Trust flag to 1.

- o Receiving of Untrusted Broadcast Root Refresh (UB-Root-Refresh):
  The node verifies if the sender of the message has an entry in the neighbor table. If not, discard the message. Else, refresh the root element and the relevant fields of the corresponding neighbor entry.

- o Receiving of Trusted Unicast Route Reply Acknowledge (TU-RREP-ACK):
  The node verifies if the sender of the message has an entry in the neighbor table and if the Neighbor-is-Valid is set to 1. If not, discard the message. Else, refresh all the relevant fields of the corresponding neighbor entry and set the Trust flag to 1.

- o Receiving of the remaining trusted messages:
  The node verifies if the sender of the message has an entry in the neighbor table and if the Trust flag is set to 1. If not, discard the message. Else, refresh all the relevant fields of the corresponding neighbor entry and set the Neighbor-is-Valid flag to 1.

2. Routing Table
   In case the sending node has a route entry in the routing table, all its information including timers, metric and sequence number will be updated and the Route-is-Valid flag is set to 1. Otherwise, a new route entry for the sending node will be created. Afterwards, a node verifies if it has a route entry for the creator of the message and undergoes the same steps as for the sending node. Finally, the node repeats this process with respect to all intermediate nodes and IP addresses included in the route address range list field of the message and the neighbors address range list, if available.

8.3. Message Generation

8.3.1. Untrusted Broadcast Route Request (UB-RREQ)

This message is generated if a node does not have a route to a desired destination. The node generates a UB-RREQ message according to Table 1. Hereby, it sets the Destination-is-Mesh-Gateway flag to 1 if the desired destination is a mesh gateway. Besides, it sets the Registration flag to 1 if the route request is part of the registration process as described in sub-section 8.1. After generating the message, the node broadcasts it and it initializes the Route_Discovery_Timeout.

8.3.2. Trusted Unicast Route Request (TU-RREQ)

After receiving a UB-RREQ message, an intermediate node, that has a route to the destination, generates this message and sends it to the next hop of that route.

8.3.3. Trusted Unicast Route Reply (TU-RREP)

Upon receiving a TU-RREQ, a destination node generates a TU-RREP. If the Mesh Gateway and the Registration flags of the TU-RREQ were set, the mesh gateway first requests the KDC-block from the key distribution center and then its replies the TU-RREP message.

8.3.4. Untrusted Unicast Route Reply (UU-RREP)

As a final response to a UB-RREQ message, an intermediate or a destination node generates a UU-RREP message. It sends this message and initializes the TU_RREP_ACK_Timeout.

8.3.5. Trusted Unicast Route Reply Acknowledge (TU-RREP-ACK)

This message is generated by an originator node when it receives a UU-RREP. This message is the last message in the trust establishment three-way handshake after which neighbors mainly communicate using trusted messages.

8.3.6. Trusted Broadcast Hello (TB-Hello)

This message is generated each time the Hello_Periodic_Broadcast_Timeout is triggered.

8.3.7. Trusted Broadcast Route Error (TB-RERR)

Upon receiving a packet for a destination the entry of which has been deleted or in case the next hop for that destination is not reachable anymore, an intermediate node generates and broadcasts a route error (TB-RERR) message.

8.3.8. Untrusted Broadcast Root Refresh (UB-Root-Refresh)

After revealing all secrets, a node generates new secrets. It then computes a new root element. Afterwards, it generates the UB-Root-Refresh message to inform neighbors about its new root element.

8.3.9. Untrusted Broadcast Key Refresh (UB-Key-Refresh)

In case a node gets compromised, the group/client transient keys are regenerated and the certificate of the compromised node is

blacklisted. In that case, the KDC informs mesh gateway nodes about the new keys by sending them a new Key-To-Use mark. A mesh gateway node generates the UB-Key-Refresh message, which is then flooded in the network.

8.4. Handling Sequence numbers

Every route table entry at every node MUST include the latest information available about the sequence number for the IP address of the destination node for which the route table entry is maintained.

At power-up every node is assigned the sequence number 1. This sequence number is increased by one every time a node sends or forwards a message. When the maximum number is reached the sequence number is reset to 1. Note that an attacker cannot misuse an old sequence number due to the security mechanisms endorsed in PASER. Sequence number is rather used to prevent message flooding and routing loops between legitimated nodes.

8.4.1. Route Reply Messages

Upon receiving a PASER route reply message (UU-RREP or TU-RREP), a node verifies if the sequence of the destination is already known and if the sequence number of the message is higher. In that case, the message is considered fresh and it will be further processed. In case the sequence number of the message is smaller than the one in the route entry, the node verifies if the difference between both numbers is higher than (2^31-1), where a sequence number has a 32 bits length. In case the difference is higher, the message is considered fresh, else the message is discarded.

8.4.2. Remaining PASER messages

Upon receiving a PASER message other than a route reply message, a node verifies if the sequence of the originator is already known and if the sequence number of the message is higher. In that case, the message is considered fresh and will be further processed. In case the sequence number of the message is smaller than the one in the route entry. The node verifies if the difference between both numbers is higher than (2^31-1) where a sequence number has a 32 bits length. In case the difference is higher, the message is considered fresh, else the message is discarded.

In case the sequence number of the originator is not known, the message is considered fresh.

In case the sequence number of the originator equals the number stored in the corresponding route entry, the node verifies the sequence number of the sender. If the sender itself is the originator, the message is discarded, else the message is considered fresh if and only if the sequence number of the sender is higher than the sequence number in the corresponding neighbor entry. In case the sequence number of the sender is not known, the message is considered fresh. This mechanism is not necessary in route reply messages, since these messages are sent over a selected route. The latter is discovered in the route request phase.

8.5. Message Processing

8.5.1. Untrusted Messages

After receiving an untrusted message, a node undergoes the following steps in the given order.

1. Verify from the timestamp and the sequence number(s) (see section 8.4) that the message is fresh. If not, discard the message.

2. Verify using geographical leashes if the neighbor / sender of the message is in the transmission range. If not, discard the message.

3. Verify if the key number equals the one the node is using. If not, send UB-Key-Refresh and discard the message.

4. Verify the authenticity of the message by verifying its digital signature. If the signature is not valid, discard the message.

5. Update routing and neighbor tables according to sub-section 8.2.

6. Depending on the message type, execute the following:

   - UB-RREQ: Verify if the node itself is the desired destination. In that case, reply with UU-RREP and initialize TU_RREP_ACK_Timeout. If the node itself is not the destination but it has a valid route to the destination, generate and send TU-RREQ to the next hop. If the node does not have a route to the destination, update and forward the UB-RREQ.

   - UU-RREP: Reply with TU-RREP-ACK. Verify if the node itself is the destination. In that case, delete Route_Discovery_Timeout. Else, verify if the next hop

towards the originator node is trusted. If not, update and forward UU-RREP, else generate and send a TU_RREP.

- o UB-Key-Refresh: verify if the key number in the key refresh message is higher than the one the node is currently using. If not, discard the message. Else, forward the message, delete PASER tables and start a registration process again.

8.5.2. Trusted Messages

1. Verify from the sequence number(s) (see section 8.4) that the message is fresh. If not, discard the message.

2. Verify using geographical leashes if the neighbor / sender of the message is in the transmission range. If not, discard the message.

3. Verify if the key number equals the one the node using. If not, send UB-Key-Refresh and discard the message.

4. Verify if the sender is a trusted neighbor. Else, discard the message.

5. In case of a TB-HELLO message, verify if the node itself is listed in the neighbor address range list field of the message, if not, discard the message.

6. Verify if the initialization vector part of the secret is higher than the one stored in the neighbor table. If not, discard the message.

7. Verify the integrity of the message by verifying the hash value using GTK. If the value is not valid, discard the message.

8. Compute from the authentication path and the secret the root element and compare it with the root of the sender in the neighbor table. If these are not equal, discard the message.

9. Update neighbor and routing tables as described in section 8.2.

10. Save the value of the initialization vector part of the secret in the corresponding field of the neighbor table.

11. Depending on the message type, execute the following:

    - o TU RREQ: verify if the node itself is the destination. In that case, reply with TU-RREP. If not and if the route to

the destination is known and the next hop is trusted and valid, then update and send the TU-RREQ to the next hop. Else, generates and send TB-RERR or undergo the local repair functionality, if activated.

- o TU-RREP: verify if the node itself is the destination. In that case, delete Route_Discovery_Timeout. Else, verify if there is a route entry to the destination and if the next hop is a trusted and valid neighbor. In that case, update and forward TU-RREP. Else, if the next hop is untrusted and valid, send UU-RREP. Else, generate and send TB-RERR or undergo the local repair functionality, if activated.

- o TU-RREP-ACK: delete TU_RREP_ACK_Timeout.

- o TB-RERR: verify whether the sequence number of each unreachable node included in the unreachable destination list field is fresh, i.e., it is higher or equal the sequence number stored in the routing table entry, if already known. If the sequence number is not known it is considered fresh. Disregard nodes with outdated sequence number. Verify if the sender of the message is the next hop to the remaining unreachable nodes. Invalidate the routing entries of those nodes for which this requirement is met. Create a new unreachable destination list comprising these nodes. Update and rebroadcast the TB-RERR message.

8.6. Local Repair

The local repair mechanism described in [9] is adopted in PASER.

8.7. Buffering of Packets to unknown destination

Data packets waiting for a route to be established (i.e., waiting for a route reply) SHOULD be buffered. The buffering SHOULD be managed according to the FIFO principle (first-in, first-out). If a route discovery has been attempted the maximum times of retries and the Route_Discovery_Timeout is triggered before receiving any route reply, all data packets destined for the corresponding destination SHOULD be dropped from the buffer. This approach is adopted from [9].

9. Security Considerations

PASER promises to achieve the following goals:

- o Node authentication: This goal is guaranteed by the digital signature in untrusted messages (including revocation list messages) and by the symmetric authentication mechanism in trusted messages - PASER is robust against impersonation and man-in-the-middle attacks.

- o Message freshness and integrity: The freshness goal is provided by the sequence number included in each message, the nonce parameter during the registration process, the timestamp in untrusted messages and the secrets in trusted messages. The integrity is achieved by the digital signature in untrusted messages and by the keyed-hash value in trusted messages - PASER is robust against replay and tempering attacks.

- o Neighbor transmission authentication: Provided position information is not falsified, PASER guarantees to a large extent that node's neighbors are always in that node transmission range. This goal is provided by the fault tolerant distance awareness between new neighbors (geographical leashes) combined with the achievement of the node authentication goal. - PASER is robust against the wormhole attack.

PASER does not fulfill the data confidentiality goal. This goal should be guaranteed by another protocol, if necessary. Only vital goals necessary to secure the routing process against external attackers are addressed by PASER. Otherwise, running other security protocols in parallel with PASER will cause an accumulation of security technologies and redundant goals resulting in huge consumption of resources.

PASER does not protect against internal malicious nodes, i.e., nodes that do not stick to the protocol behavior. PASER rather offers proactive security against none-authorized nodes and excludes compromised nodes from the network. Protecting the network against internal malicious nodes is more a reactive security concern.

10. IANA Considerations

PASER defines a "Type" field for its messages. This document requires IANA to assign the following numbers for this Type field:

Message Type	Value
Untrusted Broadcast Route Request (UB-RREQ)	1
Untrusted Unicast Route Reply (UU-RREP)	2
Trusted Unicast Route Reply Acknowledge (TU-RREP-ACK)	3

11. Conclusions

PASER is a secure and efficient position aware hierarchical routing protocol for wireless mesh networks. From a security perspective, PASER features a hybrid scheme to secure the routing process. A concise combination of digital signature, hash tree authentication scheme and keyed-hash function characterizes this protocol. Another key feature of PASER is the integration of nodes' positions in the route discovery, allowing an advanced network management while mitigating a wider range of attacks. Apart from that, to address the problem of node compromise, PASER endorses a key revocation scheme to efficiently exclude those nodes. Summing up, PASER aims to achieve a reasonable trade-off between security and performance.

12. References

12.1. Normative References

[1] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] PASER: Position Aware Secure and Efficient Mesh Routing Protocol. [Online]. Available: www.paser.info

[3] M. Sbeiti, J. Pojda, and C. Wietfeld, "Performance Evaluation of PASER - an Efficient Secure Route Discovery Approach for Wireless Mesh Networks", in IEEE PIMRC, Sydney, Australia, Sep. 2012.

12.2. Informative References

[4] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks", IEEE Wireless Communications, vol. 14, no. 5, pp. 85-91, Oct. 2007.

[5] IEEE Standard 802.11i, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements", Jul. 2004.

[6] L. Abusalah, A. Khokhar, and M. Guizani, "A Survey of Secure Mobile Ad hoc Routing Protocols", IEEE Communications Surveys and Tutorials, vol. 10, no. 4, pp. 78-93, Fourth Quarter 2008.

[7] I. Chakeres and C. Perkins, "Dynamic MANET On-Demand (AODVv2) Routing", draft-ietf-manet-dymo-23, Oct. 2012.

[8] IEEE Standard 802.11 - 2012, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", Mar. 2012.

[9] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) routing", RFC 3561, Jul. 2003.

[10] T. Clausen and P. Jacquet, "Optimized Link State Routing (OLSR) Protocol", RFC 3626, Oct. 2003

[11] Y.-C. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks", in IEEE INFOCOM, San Francisco, CA, USA, Mar. 2003.

[12] M. Sbeiti, J. Hinker, and C. Wietfeld, "VLX: A Novel Virtual Localization Extension for Geographical Leash-based Secure Routing in Indoor Wireless Mesh Scenarios", in IEEE WiMob, Barcelona, Spain, Oct. 2012.

[13] A. Menezes, P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996, ch. 13, p. 556.

[14] E. Barker and A. Roginsky, "Transitions: Recommendation for Transitioning the Use of Cryptographic, Algorithms and Key Lengths", NIST Special Publication 800-131A, Jan. 2011.

13. Acknowledgments

The authors would like to thank Eugen Paul, Andreas Wolff, Carsten Vogel, Jonas Hinker, Jan Schroder, Sebastian Rohde and Niklas Goddemeier for their assistance by the design, development and evaluation of the PASER protocol. We also acknowledge the support of the SPIDER and AVIGLE projects. SPIDER is part of the nationwide security research program funded by the German Federal Ministry of Education and Research (BMBF) (13N10238). AVIGLE is co-funded by the German Federal State North Rhine Westphalia (MIWF) and the European Union (European Regional Development Fund: Investing In Your Future).

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Mohamad Sbeiti
Communication Networks Institute
Dortmund University of Technology
Otto-Hahn-Str. 6,
44227 Dortmund, Germany

Phone: +49-231-755-6128
Email: Mohamad.sbeiti@tu-dortmund.de

Christian Wietfeld
Communication Networks Institute
Dortmund University of Technology
Otto-Hahn-Str. 6,
44227 Dortmund, Germany

Phone: +49-231-755-4515
Email: Christian.wietfeld@tu-dortmund.de