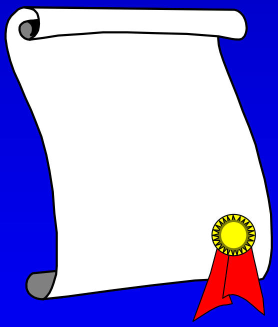
WAC 2005

Multipath Routing Protocols for Mobile Ad Hoc Networks: Security Issues and Performance Evaluation

Rosa Mavropodi
Christos Douligeris
Dept of Informatics University of Piraeus
Tel: + 30-1-4142137, Fax: 4142267
Email: {rosa, cdoulig}@unipi.gr

Contents



Introduction

Classification according to the routing mechanism

- ❑ Table driven (proactive)
- ❑ Source initiated (on demand)
- ❑ Single path
- ❑ Multipath

  a. When to initiate the routing process

     i. all path collapse

     ii. first path collapses

  b. Number of paths used

     i. only one

     ii. distribute data through several paths for a single session

  c. Complete

  d. Node-disjoint

  e. Link-disjoint

Security issues

- ❑ Node authentication
- ❑ Trust establishment
- ❑ Key agreement
- ❑ Intrusion detection
- ❑ Denial-of-Service attacks

  (impose the need of node-disjoint multipath routing protocols)

Vulnerabilities of multipath routing

- The racing phenomenon
  Intermediate nodes process
  - only the first route request query = > reduction of the discovered disjoint paths

- Impersonation and lack of authentication
  - luck of link-to-link authentication => impersonation attacks

- Invisible node
  - Man-in-the-Middle

SecMR Protocol

Secure Multipath Routing protocol (SecMR).

A novel on-demand, multipath routing protocol, secure against a bounded number of colluding malicious nodes, the SecMR discovers the complete set of the existing non-cyclic, node-disjoint paths between a source and a target node, for a given maximum hop distance.

First phase:

the *neighborhood authentication* phase, involves the asynchronous mutual authentication of neighboring nodes.

Second phase:

the *route discovery and maintenance* phase, involves the establishment and
maintenance of active routes.

This later phase consists of three algorithms,

- ✓*route request query algorithm,*
- ✓*route reply algorithm, and*
- ✓*route error algorithm.*

Neighborhood Authentication Phase

- Each node possesses public-secret keys (PK_i, SK_i), of an Elliptic Curve Cryptosystem
- A Certifying Authority CA issues a certificate $cert_i$, which, for each node, certifies its public key and also contains its unique identifier ID_i.
- The size of the identifier ID_i depends on the average network connectivity and is a relatively small number.

In periodic time intervals, each node n_i broadcasts to its one-hop neighbors a signed message including the current time and its unique identifier that is included in its certificate.

$$n_i^{\,t} = (t,\ ID_i,\ sig_i\,(t,\ ID_i),\ cert_i)$$

The duration of the time period of the neighborhood authentication phase is a system parameter and depends on the volatility of the environment.

Route Request Query (1/3)

ID_S, ID_T	are the identifiers of source (S) and target (T)
Seq	is a counter used by S for each new query
hop_{cnt}	is a counter that tracks the current number of hops
hop_{max}	is the maximum allowed hop distance
$E_{PK(T)}$ $(K_{S,T})$	is the encryption of the key $K_{S,T}$ with the public key PK_T of node T
$RouteList$	is a dynamically generated list of the intermediate nodes participating in a path between S and
$ExcludeList$	is a dynamically generated list of nodes that are excluded for a parti cular thread of the query
$NextHop$	is the list containing the nodes that are allowed to be the next hop of the particular query
$hash_{KS,T}(ID_S, ID_T, seq, hop_{max})$	is the result of a keyed hash function with the key $K_{S,T}$

$$S \xrightarrow{Q_{S,T}} T$$

$[\ ID_S,\ ID_T,\ seq,\ hop_{cnt},\ hop_{max},\ E_{PK(T)}(K_{S,T}),$

$RouteList,\ ExcludeList,\ NextHop,$

$hash_{K(S,T)}(ID_S,\ ID_T,\ seq,\ hop_{max})\ ]$

Route Request Query (2/3)

Generation of Route Request in source node

1. Initialization of the route request query:

   $hop_{cnt} = 0$, $hop_{max} = MAX$,

   $RouteList = \varnothing$, $ExcludeList = \varnothing$, $NextHop = N_s$

2. Selection of random:

   $K_{S,T}$ = The secret key (security association) to be shared between S and T

3. Computation of:

   $E_{PK(T)}(K_{S,T})$ and $hash_{K(S,T)}(ID_S, ID_T, seq, hop_{max})$

4. Broadcast of query:

$$Q_{S,T} = [\ ID_S,\ ID_T,\ seq,\ hop_{cnt},\ hop_{max},\ E_{PK(T)}(K_{S,T}),\ RouteList,\ ExcludeList,\ NextHop,\ hash_{K(S,T)}(ID_S, ID_T, seq, hop_{max})\ ]$$

Route Request Query (3/3)

```
1) If (hash(Q_{S,T}) ∈ RouteTable(n_i))
                    /* Drop duplicates of the particular request query thread */
OR((RouteList ∩ ExcludeList ≠ ∅) OR (RouteList ∩ NextHop ≠ ∅) OR (ExcludeList ∩ NextHop ≠ ∅))
                    /* Drop the query if a node identifier belongs to more than one list */
OR ((ID_i ∉ NextHop) OR (LastElement(RouteList) ∉ N_i)
                    /* Drop the query if the previous node is not a neighbor of */
   then
          DROP(Q_{S,T})
   else{

2) add(hash(Q_{S,T}), RouteTable(n_i))
                    /* n_i marks the specific route request query as processed */
3) If (ID_i = ID_T) then REPLY(Q_{S,T})
                    /* If n_i is the target, execute the route reply algorithm and exit */
   else {
4) hop_cnt = hop_cnt + 1
5) If (hop_cnt > hop_max)/* Drop the query if it exceeds the maximum allowed hop-distance */
  then
          DROP(Q_{S,T})
  else{
6) RouteList = RouteList + ID_i
                    /* Node n_i adds itself to the RouteList */
7) ExcludeList = ExcludeList + (NextHop - ID_i)
                    /* Node n_i excludes the rest of the neighbors of the previous
                    node, from this particular thread of the route request query */
8) NextHop = N_i - (N_i ∩ RouteList) - (N_i ∩ ExcludeList)
                    /* The allowed next hops of this query thread are the neighbors of n_i
                    unless they already belong to the route list or the exclude list */
9) Update the query with the new values and broadcast it

          Q_{S,T} =[ ID_S, ID_T, seq, hop_cnt, hop_max, E_{PK(T)}(K_{S,T}), RouteList,ExcludeList, NextHop,
          hash_{K(S,T)}(ID_S, ID_T,    seq, hop_max)] } }}
```

Route Reply

When the target node T receives a thread of a route request query $Q_{S,T/i}$, it decrypts $E_{PKT}(K_{S,T})$, obtains the key $K_{S,T}$ and checks the validity of the included keyed hash value.

Then, it waits for a certain amount of time in order to receive any other threads of the same route request query coming from different paths. The keyed hash-value of each thread is also checked.

Then, the target node T constructs the maximum set of node-disjoint paths M.

For each $RouteList_j \in M$, node T constructs and broadcasts a route reply message as:

$$R_{S,T/j} = [\ ID_S,\ ID_T,\ seq,\ RouteList_j,\ sig_i(ID_S,\ ID_T,\ seq,\ hash_{K(S,T)}(ID_S,\ ID_T,\ seq,\ RouteList_j)\ ]$$

Route Error

If a node n_i realizes during neighbourhood authentication at time $t+1$ that an established link with a neighbouring node n_j during time t is now broken, then node n_i broadcasts a route error message for any active route coming through n_i, that is affected due to the destruction of the link (ni, nj).

The route error message is digitally signed by the node n_i.

If the error messages are not signed, malicious nodes might flood the network with fake error messages even for routes that they do not participate in, and in this way disable communication. The route error message is of the form:

$$E_{S,T} = [\ ID_S,\ ID_T,\ seq,\ ID_i,\ RouteList,\ sig_i(ID_S,\ ID_T,\ seq,\ ID_i,\ RouteList)\ ]$$

Analysis (1/3)

- *End to end route authentication*

  The route request is end-to-end authenticated with the security association $K_{S,T}$ that is exchanged. The keyed hash-value $hash_{KS,T}$ $(ID_S, ID_T, seq, hop_{max})$ included in the initial query allows the target node to authenticate the request query.

- *Link-to-link route authentication*

  The links of a routing path are also authenticated indirectly, due to the neighborhood authentication phase of the protocol.

- *End to end route integrity*

  Each route reply message includes a keyed hash-value $hash_{KS,T}$ $(ID_S, ID_T, seq, RouteList_j)$. Thus, if the routing path $p_j = (ID_S, RouteList_j, ID_T)$ has been altered, then the verification of the keyed hash-value will fail at node *S* and the fake path will not be used.

Analysis (2/3)

- *Protection against malicious collaborating nodes*

  By using k node-disjoint paths of communication, an adversary should compromise at least k nodes - and more particularly at least one node in each route - in order to control the communication.

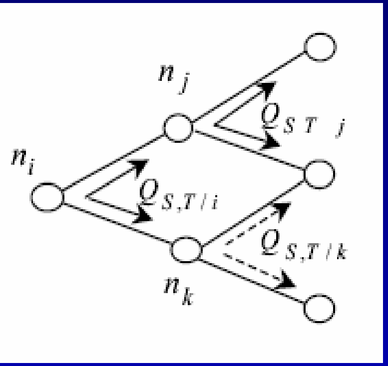
  According to the operation mode, SecMR offers different levels of protection.

  - ✓ In parallel mode, the protocol is resilient against $k - 1$ collaborating malicious nodes.
  - ✓ In single operation mode the adversary can disrupt communication by compromising only the active path. The time required to activate an alternative path is still much less than in single-path routing protocols, but there are cases where such disruption may be critical.

- *Complete* as intermediate nodes processes all the incoming requests.

Analysis (3/3)

- Node-disjoint with the use of `RouteList`.
- Non-Cyclic with the use of the `ExcludeList` which forces the query to move only to more distant nodes of S towards T.



- Message length of the used lists:

R_{max} max route length

C node connectivity (the nodes are not related to each other)

$RouteList_{max} = R_{max}$

$ExcludeList_{max} = (R_{max}-1)*(C-1)$

$NextHop_{max} = C$

For a typical network where $R_{max}=5, c=5 \Rightarrow$

$RouteList_{max}=5$ $ExcludeList_{max}=16$ $NextHop_{max}=5$

Performance Evaluation (1/6)

Compare against two other protocols with similar characteristics.

- The SRP [19] is a multipath routing protocol which aims at this kind of protection.
  SRP uses only symmetric cryptography in an end-to-end manner, to protect the integrity of the route discovery.

- Multipath[4] is a secure multipath routing protocol, based on the Ford-Fulkerson MaxFlow algorithm.

Performance Evaluation (2/6)

Comparison against two other protocols with similar characteristics.

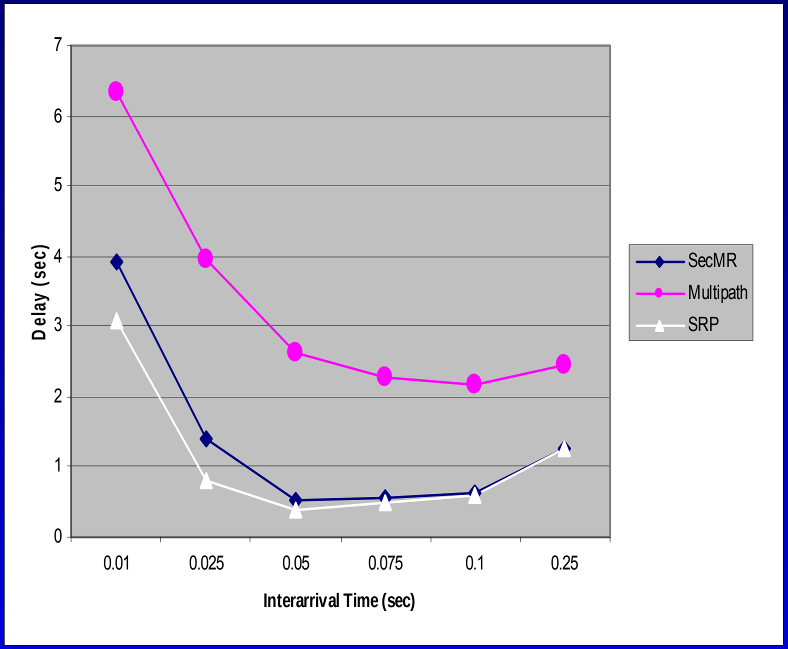
Characteristics	Protocols			Vulnerabilities (derived from the luck of this characteristic)
	SecMR	Multipath	SRP	
end-to-end authentication	yes	yes	yes	luck of data integrity
link-to-link authentication	yes	yes	no	Impersonation, sybil attacks
complete	yes	yes	no	less discovered paths
how many requests the intermediate node processes	all	all	Only the first	racing phenomenon

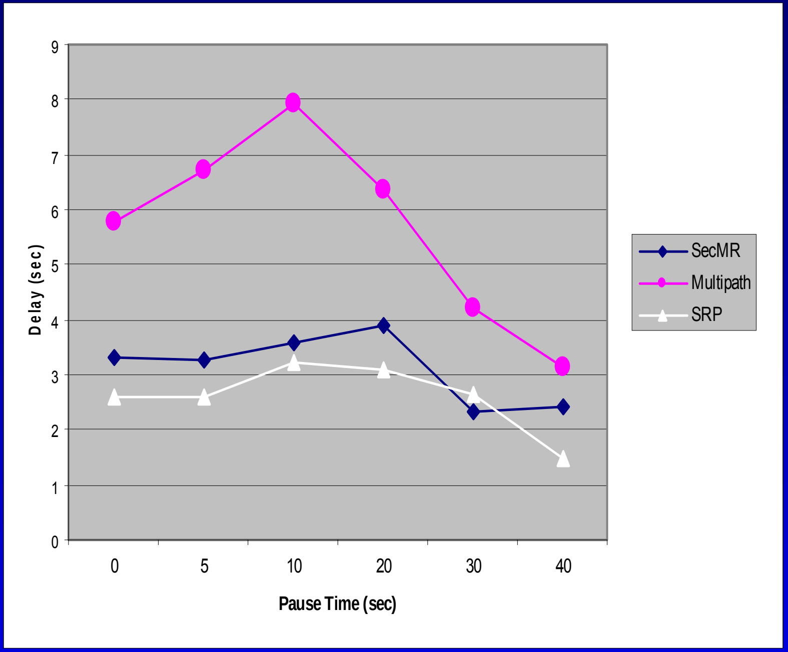
Performance (3/6)

The simulation scenario within NS-2.27 library:

- 50 hosts placed randomly within a 1000 X 1500 m^2 area
- Approximately 5 hops as neighbors with Radio propagation range of 150 meters
- Constant bit rate sources (CBR) traffic pattern with
- Channel capacity was 2 Mb/s
- Minimum and maximum speed is set to 0 and 20 m/s, respectively
- Various pause times 0, 5, 10, 20, 30, 40
- Size of the data payload was 512
- Each run executed for 350 sec of simulation time
- The sources and the destinations are randomly selected with uniform probabilities
- IEEE 802.11 Distributed Coordination Function (DCF) as the medium access control protocol
- Free space propagation model with a threshold cutoff
- Radio model locks onto a sufficiently strong signal in the presence of interfering signals, *i.e.*, radio capture

Performance (4/6)

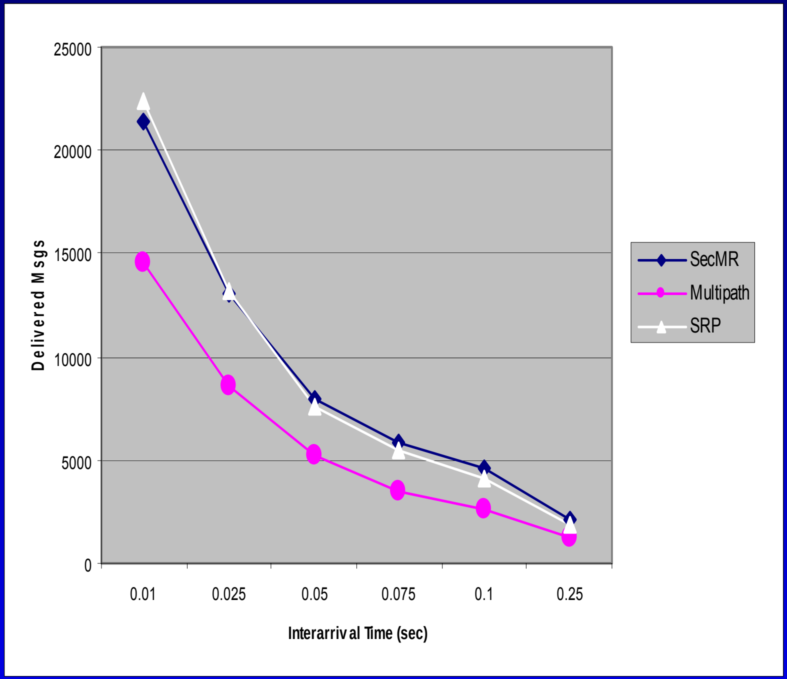




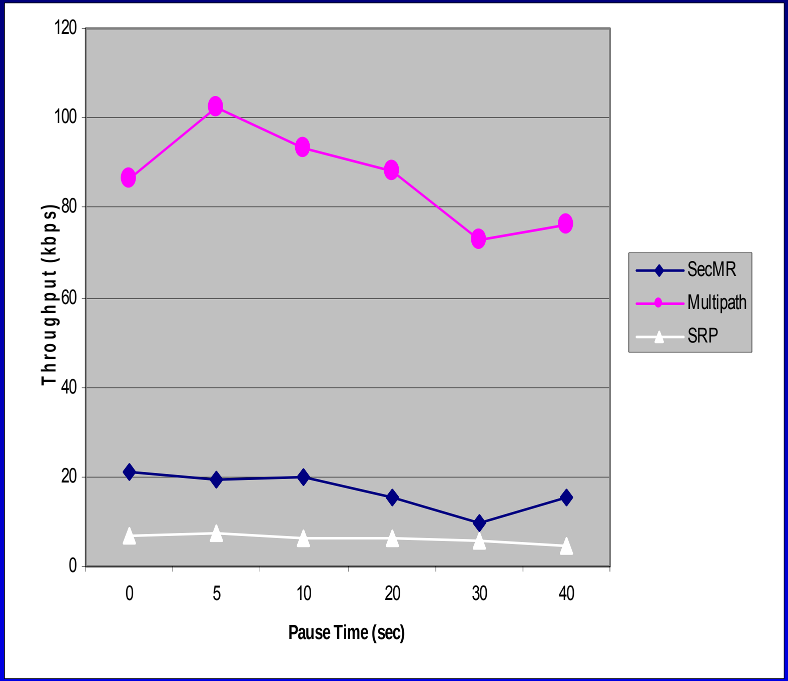
Average end-to-end data packet delay per interarrival time

Average end-to-end data packet delay per pause time

Performance (5/6)

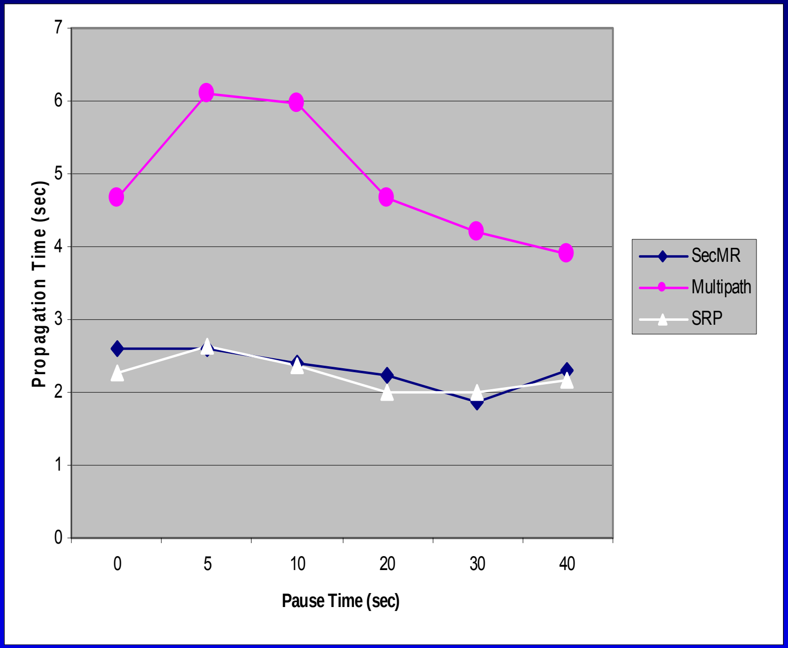


Number of delivered data packet per interarrival time

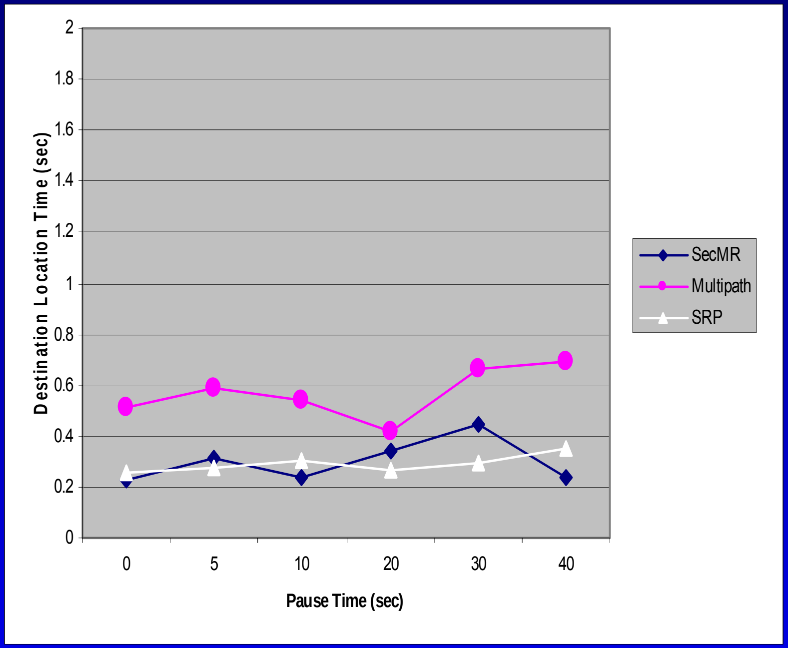


Routing throughput per pause time

Performance (6/6)



Request Propagation Time per pause time



Destination location time per pause time

Conclusions

In networks that require high security protection and present medium mobility as well as a rather high node density SecMR protocol has comparable efficiency with the SRP, while it offers an increased security level.

This is expected as Multipath[4] floods the network with route requests messages, while SecMR performs selective forward with the use of the `ExcludList`.

SRP[19] manages to perform well in networks with increased node density as it avoids discovering all the possible routes that each node could participate and in this way it converges faster, but this makes it vulnerable to distributed DoS attacks.

SeCMR combines the strong security advantageous of Multipath with a performance comparable to SRP.