Copyright Information

© 2008 Scalable Network Technologies, Inc. All rights reserved.

QualNet is a registered trademark of Scalable Network Technologies, Inc.

All other trademarks and trade names used are property of their respective companies.

Scalable Network Technologies, Inc.
6100 Center Drive, Suite 1250
Los Angeles, CA 90045

Phone: 310-338-3318
Fax: 310-338-7213

http://www.scalable-networks.com
http://www.qualnet.com
Table of Contents

Network Security - Overview .................................................................................................................... 1
Adversary Model ..................................................................................................................................... 4
Anonymous On-demand Routing (ANODR) Protocol .......................................................................... 12
Credential Model: IFF Certificate .................................................................................................... 17
ISAKMP Model .................................................................................................................................... 21
Secure Neighbor Model ..................................................................................................................... 33
WEP and CCMP Models .................................................................................................................... 41
Network Security - Overview

In a large-scale network with heterogeneous network components, Network Security becomes an essential design dimension after mobile components and wireless components are introduced into the network, for the following reasons:

- **Vulnerability of channels.** Messages can be eavesdropped upon and fake messages can be injected into the network. In wireless networks, this can even be done without having physical access to network components.

- **Vulnerability of nodes.** Nodes are always susceptible to being compromised. In mobile networks, there is no clear line of defense. Since network nodes do not usually reside in physically protected places such as locked rooms, they can more easily be captured and fall under the control of an adversary.

- **Unstable infrastructure support.** In mobile networks, nodes are capable of operating independently of any fixed infrastructure. This makes classic security solutions based on online authorities, unsuitable.

- **Highly dynamic network states.** In mobile wireless networks, the permanent changes of network states require sophisticated network protocols, the security of which is an additional challenge. A particular difficulty is that incorrect network information can be generated by either adversarial attacks, or as the result of highly dynamic network conditions, and it is hard to distinguish between the two cases.

- In the past, the challenge of implementing net-centric network security schemes to protect the network facing these vulnerabilities and network dynamics has gone unanswered. Today, QualNet provides a readily-available simulation and emulation tool that can assess any network security protocols and evaluate their performance in experimental scenarios. A QualNet assessment can save a great deal of cost, and in many cases unrecoverable casualty loss, by reducing deployment expenses and fixing flaws or failures in proposed network security plans.

**Traditional Security Approach**

Traditional security methodologies use analytical models to prove the effectiveness of a security proposal, rather than using network protocol models. Many times, formal crypto-analysis or textual arguments are used to justify the validity and quality of a security proposal. Some security proposals validated by this approach have never been implemented and tested. Mathematical formulas and various crypto-analytic methods are the basis of analytical models.

**QualNet’s NetSec Approach**

Figure 1 shows an overview of QualNet’s NetSec approach to scalable Network Security.
In QualNet’s NetSec, the following network security supports are currently available in a scalable network scenario:

**Adversary Model**

The Adversary Model (also known as: Threat Model, Attack Model, and Penetration Model) comprises:

- Active Threat, in which an adversary changes the network state, for example, by injecting packets into the network. QualNet 4.5 implements wormhole attack where the adversary disrupts ad hoc routing protocols using higher-bandwidth and lower-latency links.
- Passive (or Eavesdrop) Threat, in which wireless traffic is intercepted by an eavesdropping entity.

**Confidentiality/Privacy and Data Integrity**

At the network layer, QualNet provides supports for IPsec, which is designed to provide a cryptographically based security for IP. It provides the following security services:

1. Access Control
2. Connectionless Integrity
3. Data Origin Authentication
4. Partial Sequential Integrity
5. Confidentiality
6. Traffic Flow Confidentiality

Security is as weak as the weakest unprotected point.

**FIGURE 1. Network Security Modules Permeate the Entire IP Protocol Stack**

Network Security - Overview
These services are provided at the Network Layer and work to protect IP and upper layer protocols. The service is provided using Encapsulation Security Payload (ESP) protocol, and through the use of cryptographic key management procedures such as ISAKMP.

At the link layer, QualNet provides support for WEP and its secured successor CCMP. Wired Equivalent Privacy (WEP) is a MAC layer security protocol intended to provide security for the wireless LAN equivalent to the security provided in a wired LAN. CCMP is currently adopted by 802.11i to replace WEP. CCMP is based on the CCM of the AES encryption algorithm.

**Authentication, Trust Management and Key Management**

*Certification:* In a secured wireless network, each node must be capable of authenticating itself to its colleague network members, and vice versa. In QualNet's NetSec modeling, every network member should acquire a signed credential from an offline authority CA (Certificate Authority) prior to network operations. The credential is a certificate signed by CA's private key $SK_{CA}$, and can be verified by the well-known $PK_{CA}$, which is assumed to be cached by every network member's local storage. In a nutshell, at the time of a priori offline registration, a network member $X$ obtains $PK_{CA}$ (CA's public key) and $CERT_{X}$ ($X$'s own certificate signed by $SK_{CA}$). Certification ensures proper network authentication.

*Secure Neighbor Authentication (SNAuth):* Every mobile node authenticates its neighbors on the move. In a new neighborhood, a node initiates handshakes with its neighbors and establishes a secured connection with each of its authenticated neighbors. After secure neighbor authentication, network traffic to or from unauthenticated neighbors can be blocked, according to your security policies.

*Internet Security Association and Key Management Protocol (ISAKMP):* ISAKMP combines the security concepts of authentication, key management, and security associations to establish the required security for government, commercial, and private communications on the Internet. It defines procedures and packet formats to establish, negotiate, modify and delete a pair-wise secure connection which, in turn, can be instantiated by other protocols like IPsec ESP or even CCMP.

**Secure Routing**

Secure routing protocol ANODR is designed to provide a net-centric anonymous and untraceable routing scheme for mobile ad-hoc networks. It is based on the table-driven, on-demand routing protocol AODV. Any network scenario using AODV can use ANODR instead. Anonymity issues are critical for tactical scenarios, as allowing adversaries to trace network routes and infer the motion pattern of nodes at the end of those routes may pose serious threats to covert operations. The highly raised privacy demand poses challenging constraints on routing and data forwarding. In ANODR, identity-free routing and privacy-preserving techniques are used to confuse adversarial traffic analysts. The protocol design implements mobile anonymity and data confidentiality.
Adversary Model

The Adversary Model (also known as Threat Model, Attack Model, and Penetration Model) comprises Active Threat and Passive Threat.

**Active Threat (Wormhole attack)**

Compared to jamming, wormhole attack is more covert in nature and harder to detect. The term “wormhole” refers to an adversary carrying information and traveling faster than anyone else, thus the adversary is capable of launching unusual timing attacks. While physical wormholes do not exist, communication wormholes do exist, because adversaries can forward packets faster than regular nodes that require a queuing delay, transmission delay, and MAC contention delay.

A wormhole attacker tunnels messages received in one location in the network over a low-latency high-bandwidth link and replays them in a different location. This typically requires at least two adversarial devices colluding to relay packets along a fast channel available only to the attackers, so that can disrupt multi-hop ad hoc routing. In the presence of wormholes, the attacking nodes can selectively let routing control messages get through. Then, the wormhole link has a higher probability of being chosen as part of multi-hop routes due to its excellent packet delivery capability. Once the attacking nodes know they are en route, they can launch a *black hole* attack to drop all data packets, or a *gray hole* attack to selectively drop some critical packets.

In practice, single-hop wormholes (i.e., wormholes with both ends in the one-hop transmission range of the victim network), are typically ineffective because the wormholes cannot gain any timing advantage because of the science of physics. Recommended physical length of a wormhole link is between 1.2R and 2R where R is the nominal one-hop transmission range of the victim network. Such a wormhole link can gain significant timing advantage over a multi-hop forwarding path in the victim network. Moreover, victim network's turnaround time at the physical layer and the link layer must be properly estimated. QualNet provides two configuration parameters, WORMHOLE-VICTIM-COUNT-TURNAROUND-TIME and WORMHOLE-VICTIM-TURNAROUND-TIME, for the user to specify such delay. In IEEE 802.11 standard, this turnaround time includes all delays between the time an 802.11 receiver receives RF signals and the time the same 802.11 device finishes transmitting the corresponding response. Typically, the turnaround time includes RxRFDelay (receiving radio signals and analog-digital conversion), RxPHYDelay (decoding, de-interleaving, descrambling), MAC processing delay, TxPHYDelay (scrambling, interleaving, encoding) and TxRFDelay (digital-analog conversion and transmitting radio signals). A secure version of any network protocol must also count cryptographic delays to implement message's data origin authentication.

A wormhole link may work in different modes of operation:

- **Transparent Mode** as external adversary: Wormhole devices are not regular network members. However, to make wormhole attack work, the adversary must be able to intercept legitimate wireless messages (assuming the wormhole attackers can thwart low-probability-interception mechanisms). Messages are covertly intercepted at one location and replayed at other locations while regular network members do not know the existence of wormhole devices. In other words, the existence of the wormhole devices is transparent to regular network nodes. A corresponding implementation uses layer-1 devices in the victim network and layer-2 devices in the attacking network to implement the wormhole devices.

- **Participant Mode** as internal adversary: Wormhole devices are regular network members. They are compromised nodes with legitimate network addresses like IP addresses and MAC addresses. A corresponding implementation uses layer-3 devices to implement the wormhole devices. Because
wormholes working in the transparent mode already significantly thwart victim network’s routing functions, the participant mode is currently not implemented due to implementation redundancy.

**Passive Threat (Eavesdrop)**

Wireless traffic can be intercepted by any eavesdropping entity in the network, particularly, as mobile wireless nodes of the adversary. Each eavesdropper has an IP protocol stack. If needed, it can be an internal adversary/compromised node to participate in network functions. The eavesdropped packets are output to a file, the format of which is described in the Statistics section.

**Command Line Configuration**

This section explains how to configure Wormhole and Eavesdrop via the command line.

**Wormhole Configuration**

Wormhole can be configured for a subnet using the following syntax:

```
[<Subnet>] MAC-PROTOCOL MAC-WORMHOLE
```

Wormhole configuration parameters are described in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[subnet] WORMHOLE-MODE [THRESHOLD</td>
<td>ALLPASS</td>
</tr>
<tr>
<td></td>
<td>THRESHOLD : Wormhole drops any packet with size greater than or equal to the threshold value.</td>
</tr>
<tr>
<td></td>
<td>ALLPASS : Wormhole passes all packets irrespective of their size.</td>
</tr>
<tr>
<td></td>
<td>ALLEDROP : Wormhole drops all packets irrespective of their size.</td>
</tr>
<tr>
<td></td>
<td>The default values is THRESHOLD.</td>
</tr>
<tr>
<td>[subnet] WORMHOLE-THRESHOLD &lt;threshold&gt;</td>
<td>Specifies the threshold value for Wormhole.</td>
</tr>
<tr>
<td></td>
<td>&lt;threshold&gt; is in the range 0 to 2147483647.</td>
</tr>
<tr>
<td>Note: You need to specify this parameter only when the WORMHOLE-MODE is set as THRESHOLD. The default value is 100.</td>
<td></td>
</tr>
</tbody>
</table>

| [subnet] WORMHOLE-REPLAY-MAC-PROTOCOL <string> | Specifies the replay medium access protocol.                               |
|                                                | The default values is WORMHOLE-CSMA.                                       |

| [subnet] WORMHOLE-LINK-BANDWIDTH <bandwidth>  | Specifies the wormhole link bandwidth for Wormhole.                        |
|                                                | <bandwidth> is in the range 1 to 1000000000000000.                         |

| [subnet] WORMHOLE-PROPAGATION-DELAY <delay>   | Specifies the wormhole propagation delay.                                  |
|                                                | <delay> is in the range 1 to 1000000000000000.                            |
TABLE 1. Wormhole Configuration Parameters (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&lt;node-id&gt;</td>
<td>&lt;subnet&gt;] WORMHOLE-VICTIM-COUNT-TURNAROUND-TIME [YES</td>
</tr>
<tr>
<td>[&lt;node-id&gt;</td>
<td>&lt;subnet&gt;] WORMHOLE-VICTIM-TURNAROUND-TIME &lt;turnaround-time&gt;</td>
</tr>
</tbody>
</table>

**Eavesdrop Configuration**

To configure Eavesdrop, use the following parameter:

```
[node-id] | [subnet] EAVESDROP-ENABLED YES | NO
```

**GUI Configuration**

This section describes how to configure Wormhole and Eavesdrop using QualNet GUI.

**Wormhole Configuration**

To configure Wormhole for a subnet, go to Hierarchy > Nodes > Wireless Subnet [id] > Wireless Subnet Properties > MAC Protocol. In the Configurable Property window, set MAC Protocol to WORMHOLE, as shown in Figure 2.
Eavesdrop Configuration
To configure eavesdrop for a node, go to Hierarchy > Nodes > host # > Node configurations > Network Protocol. In the Configurable Property window, set Enable Eavesdropping? to YES, as shown in Figure 3.

FIGURE 2. Enabling Wormhole
Statistics

Wormhole Statistics

The Wormhole statistics are shown in Table 2.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frames intercepted all</td>
<td>Number of frames intercepted by the wormhole node.</td>
</tr>
<tr>
<td>Frames dropped by wormhole</td>
<td>Number of frames dropped by the wormhole link (since the frames are classified as data packets, for example, with packet size greater than a threshold).</td>
</tr>
<tr>
<td>Frames tunneled</td>
<td>Number of frames tunneled by the wormhole node. (Frames intercepted multiple times due to repetitive replay will not be tunneled.)</td>
</tr>
</tbody>
</table>
Eavesdrop Output

Eavesdrop does not print any statistics to the statistics (.stat) file. Instead a file is generated for each interface that records the eavesdropped packets. The file for an interface is named “default.eavesdrop.<interface-address>”. The output file contains the following information, which is explained in Table 3:

```
  time: ip_v ip_hl ip_tos ip_len ip_id
    flags ip_reserved ip_dont_fragment ip_more_fragments ip_fragment_offset ip_ttl
    ip_p ip_sum ip_src ip_dst
```

**Table 3. Eavesdrop Output**

<table>
<thead>
<tr>
<th>Output Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
<td></td>
</tr>
<tr>
<td>ip_v</td>
<td>IP Version 4</td>
</tr>
<tr>
<td>ip_hl</td>
<td>IP Header</td>
</tr>
<tr>
<td>ip_tos</td>
<td>IP type of services</td>
</tr>
<tr>
<td>ip_len</td>
<td>Total length of the IP header</td>
</tr>
<tr>
<td>ip_id</td>
<td>IP identification</td>
</tr>
<tr>
<td><strong>Flags</strong></td>
<td></td>
</tr>
<tr>
<td>ip_reserved</td>
<td>To distinguish SDR control packets</td>
</tr>
<tr>
<td>ip_dont_fragment</td>
<td>To handle fragmentation/offset whenever needed</td>
</tr>
<tr>
<td>ip_more_fragments</td>
<td>To handle fragmentation/offset whenever needed</td>
</tr>
<tr>
<td>ip_fragment_offset</td>
<td>To handle fragmentation/offset whenever needed</td>
</tr>
<tr>
<td>ip_ttl</td>
<td>IP time to live</td>
</tr>
<tr>
<td>ip_p</td>
<td>Transport protocol</td>
</tr>
<tr>
<td>ip_sum</td>
<td>Checksum</td>
</tr>
<tr>
<td>ip_src</td>
<td>Source IP</td>
</tr>
<tr>
<td>ip_dst</td>
<td>Destination IP</td>
</tr>
</tbody>
</table>

Sample Scenarios

**Wormhole Sample Scenario**

Sample Description

In the sample scenario shown in Figure 4, nodes 1 and 3 are connected to a wireless subnet. Nodes 5 and 6 are connected through another wireless subnet. Nodes 2 and 4 are wormhole nodes connected to a
Adversary Model

subnet. Wormhole is enabled on the subnet. One CBR application is configured from node 1 to node 6. 100 packets are sent from node 1 to node 6.

FIGURE 4. Wormhole Sample Scenario

Command Line Configuration
Include the following lines in the scenario configuration (.config) file:

```
# Nodes are placed and connected through these wireless subnets
SUBNET N8-192.0.0.0 {2 4}
SUBNET N8-192.0.1.0 {5 6}
SUBNET N8-192.0.2.0 {1 3}

# At Subnet level: Wormhole is configured as follows:
[N8-192.0.0.0] MAC-PROTOCOL MAC-WORMHOLE
[N8-192.0.0.0] WORMHOLE-MODE THRESHOLD
[N8-192.0.0.0] WORMHOLE-THRESHOLD 100
[N8-192.0.0.0] WORMHOLE-REPLAY-MAC-PROTOCOL WORMHOLE-CSMA
[N8-192.0.0.0] WORMHOLE-LINK-BANDWIDTH 100000000
[N8-192.0.0.0] WORMHOLE-PROPAGATION-DELAY 2US
WORMHOLE-VICTIM-COUNT-TURNAROUND-TIME YES
WORMHOLE-VICTIM-TURNAROUND-TIME 1MS
```

Include the following line in the application configuration (.app) file.

```
CBR 1 6 100 512 1S 1S 0 PRECEDENCE 0
```

Eavesdrop Sample Scenario

Scenario Description
In the sample scenario shown in Figure 5, nodes 1, 3 and 5 are connected to a wireless subnet. Nodes 2 and 4 are eavesdrop enabled nodes connected to a different subnet. One CBR application is configured from node 1 to node 5.
FIGURE 5. Eavesdrop Sample Scenario

Command Line Configuration
Include the following lines in the scenario configuration (.config) file:

```
# Nodes are placed and connected through these wireless subnets
SUBNET N8-192.0.0.0 {1 3 5}
SUBNET N8-192.0.1.0 {2 4}

# At Node level: Eavesdrop is enabled as follows:
[2 4] EAVESDROP-ENABLED YES
```
Anonymous On-demand Routing (ANODR) Protocol

ANonymous On-Demand Routing (ANODR) is designed to provide an anonymous and untraceable routing scheme for mobile ad-hoc networks. It is based on table-driven AODV, and therefore any QualNet simulation scenario using AODV can also use ANODR instead to implement anonymous routing.

A discussion of privacy in mobile wireless networks uses different terminology than that traditionally used for banking systems and the wired Internet. In addition to traditional ideas of privacy, mobile privacy has concerns for the mobile node's identity, location, and motion patterns.

Anonymity issues are critical for ANODR scenarios, since allowing adversaries to trace network routes and infer the motion pattern of nodes at the end of those routes may pose serious threats to covert operations. This heightened privacy demand poses challenging constraints on routing and data forwarding. ANODR allows you to protect your mobile wireless communication from being traced, and without the necessity of removing your device's battery. ANODR provides the following security services:

1. Negligibility-based anti-tracing such that signal interceptors cannot trace signal transmitters mobility pattern via wireless signal tracing (with non-negligible probability defined on the victim network's size).
2. Confidentiality and anonymity.
3. Traffic flow confidentiality.
4. Identity-free routing.
5. One-time packet contents such that any two wireless transmissions are indistinguishable with each other in regard to a cryptanalyst.

These services are provided at the Network Layer and Link Layer to protect the IP and link layer protocols.

The ANODR configuration is based on AODV parameter settings. ANODR parameters use the same terminology as AODV's parameters, except the name is changed from AODV to ANODR.

Command Line Configuration

ANODR can be configured at the global, node, subnet, or interface level using the following parameter.

```
ROUTING-PROTOCOL ANODR
```

Table 4 describes the ANODR configuration parameters.
Anonymous On-demand Routing (ANODR) Protocol

**Note:** All parameters in Table 4 are optional and can be configured at the global, node, subnet, and interface levels.

**TABLE 4. ANODR Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANODR-NET-DIAMETER</td>
<td>Specifies the maximum possible number of hops between two nodes in the network. The default value is 35.</td>
</tr>
<tr>
<td>ANODR-NODE-TRAVERSAL-TIME</td>
<td>Specifies the conservative estimate of the average one-hop traversal time for packets and includes queuing, transmission, propagation and other delays. The default value is 40MS.</td>
</tr>
<tr>
<td>ANODR-ACTIVE-ROUTE-TIMEOUT</td>
<td>Specifies the timeout time for an active route. Each time a data packet is sent, the lifetime of that route is updated to this value.</td>
</tr>
<tr>
<td>ANODR-BUFFER-MAX-PACKET</td>
<td>Maximum number of packets the message buffer of ANODR can hold. If the buffer fills up, incoming packets for the buffer will be dropped.</td>
</tr>
<tr>
<td>ANODR-BUFFER-MAX-BYTE</td>
<td>If nothing is specified, the buffer overflow is checked by the number of packets in the buffer. When a value is specified here, incoming packets are dropped if the incoming packet size + current size of the buffer exceed this value. The default value is 0.</td>
</tr>
<tr>
<td>ANODR-RREQ-RETRIES</td>
<td>The number of times a particular RREQ is present in case the corresponding RREP is not received.</td>
</tr>
</tbody>
</table>

**GUI Configuration**

To configure ANODR in QualNet GUI, go to Hierarchy > Node Configurations > Routing Protocol > Routing Policy > Routing Protocol for IPv4. In the Configurable Property window, set Routing Protocol for IPv4 to ANODR and set the ANODR parameters, as shown in Figure 6.
FIGURE 6. Configuring ANODR Routing Protocol
Statistics

Table 5 shows the statistics collected by ANODR.

**TABLE 5. ANODR Statistics**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of RREQ Initiated</td>
<td>Number of RREQ initiated for new connections.</td>
</tr>
<tr>
<td>Number of RREQ Retried</td>
<td>Number of RREQ re-initiated for existing connections.</td>
</tr>
<tr>
<td>Number of RREQ Forwarded</td>
<td>Number of RREQ forwarded as intermediate forwarder.</td>
</tr>
<tr>
<td>Number of RREQ Received</td>
<td>Number of any RREQ received.</td>
</tr>
<tr>
<td>Number of Duplicate RREQ Received</td>
<td>Number of duplicated RREQ received.</td>
</tr>
<tr>
<td>Number of RREQ Received by Dest</td>
<td>Number of RREQ received as destination.</td>
</tr>
<tr>
<td>Number of RREQ received by Dest with global</td>
<td>Number of RREQ received as destination and the RREQ is using efficient</td>
</tr>
<tr>
<td>trapdoor in symmetric key encryption</td>
<td>symmetric-key based global trapdoor.</td>
</tr>
<tr>
<td>Number of RREP Initiated as Dest</td>
<td>Number of RREP initiated.</td>
</tr>
<tr>
<td>Number of RREP Forwarded</td>
<td>Number of RREP forwarded as intermediate forwarder.</td>
</tr>
<tr>
<td>Number of RREP ACKed</td>
<td>Number of AACK initiated to ack RREP.</td>
</tr>
<tr>
<td>Number of RREP Received</td>
<td>Total Number of RREP received.</td>
</tr>
<tr>
<td>Number of RREP Received as Source</td>
<td>Number of RREP received as source.</td>
</tr>
<tr>
<td>Number of AACK Received</td>
<td>Total Number of AACK received.</td>
</tr>
<tr>
<td>Number of AACK Received</td>
<td>Number of RERR initiated.</td>
</tr>
<tr>
<td>Number of RERR Forwarded</td>
<td>Number of RERR forwarded.</td>
</tr>
<tr>
<td>Number of RERR ACKed</td>
<td>Number of AACK initiated to ack RERR.</td>
</tr>
<tr>
<td>Number of RERR Received</td>
<td>Number of RERR received.</td>
</tr>
<tr>
<td>Number of Data packets sent as Source</td>
<td>Number of data packets initiated.</td>
</tr>
<tr>
<td>Number of Data Packets Forwarded</td>
<td>Number of data packets forwarded.</td>
</tr>
<tr>
<td>Number of Data Packets Received</td>
<td>Number of data packets received.</td>
</tr>
<tr>
<td>Number of Data Packets Dropped for no route</td>
<td>Number of data packets dropped because of having no route.</td>
</tr>
<tr>
<td>Number of Data Packets Dropped for buffer</td>
<td>Number of data packet dropped because of being over the cache limit.</td>
</tr>
<tr>
<td>overflow</td>
<td></td>
</tr>
<tr>
<td>Number of times link broke</td>
<td>Number of link breakage detected.</td>
</tr>
</tbody>
</table>

Sample Scenario

*Scenario Description*

In the sample scenario shown in Figure 7, nodes 1, 2, 3, and 4 are connected through a wireless subnet. ANODR routing protocol is configured on all the nodes. Node 1 sends 100 data packets to node 3 via a CBR session.
FIGURE 7. ANODR Sample Scenario

**Command Line Configuration**
Include the following lines in the scenario configuration (.config) file:

```plaintext
# 4 nodes are placed connected through a wireless subnet
SUBNET N8-192.0.0.0 { 1 thru 4 }

# At Node level: ANDOR is configured as below:
[ 1 thru 4 ] ROUTING-PROTOCOL ANODR
[ 1 thru 4 ] ANODR-NET-DIAMETER 35
[ 1 thru 4 ] ANODR-NODE-TRAVERSAL-TIME 40MS
[ 1 thru 4 ] ANODR-ACTIVE-ROUTE TIMEOUT 5S
[ 1 thru 4 ] ANODR-BUFFER-MAX-PACKET 100
[ 1 thru 4 ] ANODR-BUFFER-MAX-BYTE 0
[ 1 thru 4 ] ANODR-RREQ-RETRIES 2
```
Credential Model: IFF Certificate

The certificate model implements credentials for the purpose of authentication, IFF (Identification of Friend and Foe), authorization, access control, accounting and auditing. In digital signature systems built on top of public key crypto systems, a signature signed by private key SK can be verified by the corresponding public key PK, and the signature cannot be forged by anyone not knowing the signing key SK.

In a secured wireless network, each node must be capable of authenticating itself to its colleague network members, and vice versa. In QualNet's NetSec modeling, every network member must acquire a signed credential from an offline authority or Certificate Authority (CA) prior to network operations. The credential is a certificate signed by the CA's private key $SK_{CA}$ and can be verified by the well-known public key $PK_{CA}$, which is assumed to be cached by every network member's local storage. In summary, at the time of a priori offline registration, network member $X$ obtains $PK_{CA}$ (CA's public key) and $CERT_X$ ($X$'s own certificate signed by $SK_{CA}$).

The certificate $CERT_X$ is in the form of $[X,pk_X,validtime]$ signed by $SK_{CA}$ where unique id $X$ is assigned to a node, $pk_X$ is the certified public key of the id $X$, and $validtime$ limits the valid period of the certificate. In QualNet, $X$ is a unique network address, like an IP address. For example, on a node having multiple network interfaces with IP addresses 11.11.11.11 and 22.22.22.22, the node must obtain two different certificates for both of its network interfaces, respectively.

This certificate modeling is provided for authentication services in the entire protocol stack. The current implementation uses a short certificate format defined by WTLS. Certificate renewal and revocation are not implemented. Distributed solutions of certificate renewal and revocation are discussed in Ubiquitous and Robust Security Architecture (URSA) and similar proposals relying on threshold cryptography. URSA proposes to distribute partial shares of the certificate signing key $SK_{CA}$ to $n$ nodes playing the role of partial CA, and $k$ out of $n$ partial CAs can produce $k$ partial certificates which combine into a full certificate (or certificate-revocation/counter-certificate). The scheme tolerates up to $k-1$ node intrusions and $n-k$ node crashes.

Command Line Configuration

The Certificate model can be enabled at the global, node, subnet, or interface level using the following parameter:

```
CERTIFICATE-ENABLED      YES
```

The default value of this parameter is NO.
Table 6 describes the Certificate model configuration parameters.

**TABLE 6. Certificate Model Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CERTIFICATE-FILE-LOG [YES</td>
<td>Specifies whether the certificate contents are logged in a file. If this parameter is set to YES, the certificate contents are logged in the file “default.certificate.&lt;interface-address&gt;”. This parameter can be configured at the global, node, subnet, and interface levels. The default value is YES.</td>
</tr>
<tr>
<td>NO]</td>
<td></td>
</tr>
</tbody>
</table>

**GUI Configuration**

To configure the Certificate model in QualNet GUI, follow these steps:

1. Go to **ConfigSettings > Network protocols > Network Protocol > Enable IFF certification?** In the Configurable Property window, set **Enable IFF certification?** to **YES**, as shown in the Figure 8.

![Figure 8](image)

**FIGURE 8. Enabling the Certificate Model**

2. To enable generation of the certificate log file, go to **ConfigSettings > Network protocols > Network Protocol > Enable IFF certification?** In the Configurable Property window, set **Do certificate file log?** to **YES**, as shown in the Figure 9.
Statistics
There are no statistics generated for the Certificate model.

Sample Scenario

Scenario Description
In the sample scenario shown in Figure 10, nodes 1, 2, 3, and 4 are connected through a wireless subnet. IFF certification is enabled on all the nodes.
FIGURE 10. Sample Scenario for Certificate Model

**Command Line Configuration**

Include the following lines in the scenario configuration (.config) file:

```
# Four nodes are placed connected through a wireless subnet
SUBNET N8-192.0.0.0 {1 thru 4}

# At Node level: IFF Certification is configured as below
[1 thru 4] CERTIFICATE-ENABLED YES
```
ISAKMP Model

Internet Security Association and Key Management Protocol (ISAKMP) provides a general framework to other security protocols for creating and maintaining Security Associations (SAs) in an Internet environment. The ISAKMP host negotiates SAs (ISAKMP SA) with other ISAKMP hosts and other security protocol and services. Use these ISAKMP SA to create their own SAs.

The SA feature coupled with authentication and key establishment allows users to choose their own security service, key exchange technique, encryption algorithm, and authentication mechanism based on their requirement with other users. For this, ISAKMP defines the general format and various payloads.

Command Line Configuration

The ISAKMP model can be configured at the global, node, subnet, or interface level specifying the following parameter in the scenario configuration (.config) file:

```
ISAKMP-SERVER       YES
```

The default value of this parameter is NO.

Table 7 shows the ISAKMP parameters which must be specified in the scenario configuration file.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&lt;node-id&gt;</td>
<td>&lt;interface-address&gt;] ISAKMP-CONFIG-FILE &lt;filename&gt;</td>
</tr>
<tr>
<td>ISAKMP-PHASE-1-START-TIME &lt;time&gt;</td>
<td>Specifies the time (in seconds) after the initialization, when Phase 1 negotiation starts. This is an optional parameter and can be specified at the global, node, subnet or interface level. The default value is 30S.</td>
</tr>
<tr>
<td>ISAKMP-ENABLE-IPSEC [YES</td>
<td>NO]</td>
</tr>
</tbody>
</table>

Format of the ISAKMP Configuration File

ISAKMP parameters can be configured in the ISAKMP configuration (.isakmp) file for a node interface using the following steps.
ISAKMP Model

1. Specify all the peer servers, with whom this interface will negotiate the ISAKMP exchanges, using the following syntax.

   NODE node-interface-ipv4-address
   PEER peer-ipv4-address <node-peer configuration>

   where <node-peer-configuration> is the name of the section under which the phase-1 and phase-2 configuration parameters are specified for this peer.

   **Example 1:** Interface 192.168.3.1 of Node 3 has interface 192.168.3.2 of Node 4 as its peer server.

   NODE 192.168.3.1
   PEER 192.168.3.2 3-4-Config

   **Example 2:** Interface 192.168.3.1 of Node 3 has interface 192.168.3.2 of Node 4 and interface 192.168.3.3 of Node 5 as its peer servers.

   NODE 192.168.3.1
   PEER 192.168.3.2 3-4-Config
   PEER 192.168.3.3 3-5-Config

2. Specify PHASE 1 and PHASE 2 configuration parameters using the following syntax:

   **Note:** All the parameters are mandatory and need to be specified in the given order.

   PHASE 1
   DOI <Domain-of-Interpretation>
   EXCHANGE_TYPE <exchange-type>
   FLAGS <flags e.g. ACE, CE, AE, AC, A, C, E, NONE>
   TRANSFORMS <transform 1> [<transform 2>]...

   PHASE 2
   LOCAL-ID-TYPE <Local-id-type given by RFC>
   LOCAL-NETWORK <Local-network-address>
   LOCAL-NETMASK <Local-network-mask>
   REMOTE-ID-TYPE <Remote-id-type given by RFC>
   REMOTE-NETWORK <Remote-network-address>
   REMOTE-NETMASK <Remote-network-mask>
   UPPER-LAYER-PROTOCOL <TCP or UDP>
   DOI <Domain-of-Interpretation>
   EXCHANGE_TYPE <exchange-type>
   FLAGS <flags e.g. ACE, CE, AE, AC, A, C, E, NONE>
   PROPOSALS <Proposal-1> [Proposal-2]...
Example:

3-4-Config

PHASE 1
DOI ISAKMP.DOI
EXCHANGE_TYPE EXCH_AGGR
FLAGS AC
TRANSFORMS 3DES-SHA

PHASE 2
LOCAL-ID-TYPE IPV4_ADDR_SUBNET
LOCAL-NETWORK 192.168.5.0

LOCAL-NETMASK 255.255.255.0
REMOTE-ID-TYPE IPV4_ADDR_SUBNET
REMOTE-NETWORK 192.168.1.0
REMOTE-NETMASK 255.255.255.0
UPPER-LAYER-PROTOCOL UDP
DOI IPSec.DOI
EXCHANGE_TYPE EXCH_IDENT
FLAGS AC
PROPOSALS ESP-DES-MD5-PFS AH-MD5-PFS

3. Specify the proposals for Phase 2 configuration using the following syntax:

PROTOCOLS <Protocol 1> [<Protocol 2>]

Multiple proposals could be specified for Phase 2 configuration. But Protocols configuration should be
associated with each proposal configuration.

Example 1:

ESP-DES-MD5-PFS
PROTOCOLS ESP-DES-MD5

Example 2:

ESP-DES-MD5-PFS
PROTOCOLS ESP-DES-MD5
AH-MD5-PFS
PROTOCOLS AH-MD5

4. Specify the Protocols configuration using the following syntax:

PROTOCOL_ID protocol-id
TRANSFORMS <Transform 1> [<Transform 2>]

Example 1:

ESP-DES-MD5
PROTOCOL_ID ESP
TRANSFORMS ESP-DES-MD5-PFS-XF
Example 2:

AH-MD5

TRANSFORM_ID AH
TRANSFORMS AH-MD5-PFS-XF

5. Specify the Phase-1 Transforms configuration using the following syntax:

| TRANSFORM_NAME | <Transform Name as specified in Phase-1 config> |
| TRANSFORM_ID   | <Transform Id as specified in RFC 2407> |
| ENCRYPTION_ALGORITHM | <Name of encryption-algorithm or DEFAULT> |
| HASH_ALGORITHM   | <Name of hash-algorithm or DEFAULT> |
| AUTHENTICATION_METHOD | <type of auth-method or DEFAULT> |
| GROUP_DESCRIPTION | <name of group-description or DEFAULT> |
| Life            | <life time of this transform, in minutes> |

Example 1:

TRANSFORM_NAME 3DES-SHA
TRANSFORM_ID KEY_IKE
ENCRYPTION_ALGORITHM 3DES-CBC
HASH_ALGORITHM SHA
AUTHENTICATION_METHOD RSA_SIG
GROUP_DESCRIPTION MODP_1024
Life 10

Example 2:

TRANSFORM_NAME DES-MD5
TRANSFORM_ID KEY_IKE
ENCRYPTION_ALGORITHM DES-CBC

HASH_ALGORITHM MD5
AUTHENTICATION_METHOD PRE_SHARED
GROUP_DESCRIPTION MODP_1024
Life 7

6. Specify the Phase-2 Transforms configuration using the following syntax:

| TRANSFORM_NAME | <Transform Name as specified in Protocol config> |
| TRANSFORM_ID   | <specified by RFC> |
| ENCAPSULATION_MODE | <TUNNEL, TRANSPORT or DEFAULT> |
| GROUP_DESCRIPTION | <group-description or DEFAULT> |
| AUTHENTICATION_ALGORITHM | <auth-algo or DEFAULT> |
| Life            | |

Example 1:

TRANSFORM_NAME ESP-DES-MD5-PFS-XF
TRANSFORM_ID ESP_DES
ENCAPSLULATION_MODE TUNNEL
GROUP_DESCRIPTION MODP_1024
AUTHENTICATION_ALGORITHM HMAC-MD5
Life 10
Example 2:
TRANSFORM_NAME AH-MD5-PFS-XF
TRANSFORM_ID AH_MD5
ENCAPSULATION_MODE TUNNEL
GROUP_DESCRIPTION MODP_1024
AUTHENTICATION_ALGORITHM HMAC-MD5
Life 15

Table 8 shows the possible values of different parameters.

**TABLE 8. ISAKMP Parameter Values**

<table>
<thead>
<tr>
<th>Encryption Algorithm</th>
<th>Hash Algorithm</th>
<th>Authentication Method</th>
<th>Group Description</th>
<th>Authentication Algorithm</th>
<th>Encapsulation Mode</th>
<th>Exchange Type</th>
<th>LOCAL-ID-TYPE</th>
<th>REMOTE-ID-TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES-CBC</td>
<td>MD5</td>
<td>RSA_SIG</td>
<td>MODP_768</td>
<td>HMAC-MD5</td>
<td>TRANSPORT</td>
<td>EXCH_BASE</td>
<td>IPV4_ADDR</td>
<td></td>
</tr>
<tr>
<td>*3DES-CBC</td>
<td>SHA</td>
<td>PRE_SHARED</td>
<td>MODP_1024</td>
<td>HMAC-SHA1</td>
<td>TUNNEL</td>
<td>EXCH_IDENT</td>
<td>IPV4_ADDR_SUBNET</td>
<td></td>
</tr>
<tr>
<td>*SIMPLE</td>
<td>DEFAULT</td>
<td>DEFAULT</td>
<td>DEFAULT</td>
<td>HMAC-MD5-96</td>
<td>DEFAULT</td>
<td>EXCH_AUTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*BLOWFISH</td>
<td>H-CBC</td>
<td>DEFAULT</td>
<td>DEFAULT</td>
<td>HMAC-SHA1-1-96</td>
<td></td>
<td>EXCH_AGGR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NULL</td>
<td></td>
<td>*NULL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*DEFAULT</td>
<td></td>
<td>*DEFAULT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes
1. Values beginning with * are not supported by ESP Implementation.
2. Values for other parameters are as defined in RFC 2407.

**GUI Configuration**

To configure ISAKMP for an interface, perform the following steps:

1. Go to **Hierarchy > Nodes > host [id] > Interface > Interface [id] > Interface configurations > Network Protocol** > **Enable ISAKMP?** In the configurable property window, set **Enable ISAKMP?** to **YES**, as shown in Figure 11.
FIGURE 11. Enabling ISAKMP for an Interface

2. Go to Hierarchy > Nodes > host [id] > Interface > Interface [id] > Interface configurations > Network Protocol > Enable ISAKMP? In the configurable property window, set Enable ISAKMP? to YES, as shown in Figure 11. In the configurable property window, set the ISAKMP parameters, as shown in Figure 12.
FIGURE 12. Configuring ISAKMP Parameters
ISAKMP Model

Statistics

Table 9 shows the statistics collected by ISAKMP.

**TABLE 9. ISAKMP Statistics**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Base Exchange</td>
<td>Number of base exchanges performed.</td>
</tr>
<tr>
<td>Total Number of Identity Protection Exchange</td>
<td>Number of identity protection exchanges performed.</td>
</tr>
<tr>
<td>Total Number of Authentication Only Exchange</td>
<td>Number of authentication only exchanges performed.</td>
</tr>
<tr>
<td>Total Number of Aggressive Exchange</td>
<td>Number of aggressive exchanges performed.</td>
</tr>
<tr>
<td>Total Number of Information Exchange Send</td>
<td>Number of informational exchanges sent.</td>
</tr>
<tr>
<td>Total Number of Information Exchange Receive</td>
<td>Number of informational exchanges received.</td>
</tr>
<tr>
<td>Total Number of Exchange Dropped</td>
<td>Number of exchanges dropped.</td>
</tr>
<tr>
<td>Total Number of SA Payload Send</td>
<td>Number of SA payload messages sent.</td>
</tr>
<tr>
<td>Total Number of SA Payload Rcv</td>
<td>Number of SA payload messages Received.</td>
</tr>
<tr>
<td>Total Number of Nonce Payload Send</td>
<td>Number of nonce payload messages sent.</td>
</tr>
<tr>
<td>Total Number of Nonce Payload Rcv</td>
<td>Number of nonce payload messages Received.</td>
</tr>
<tr>
<td>Total Number of Key Exchange Payload Send</td>
<td>Number of key exchange payload messages sent.</td>
</tr>
<tr>
<td>Total Number of Key Exchange Payload Rcv</td>
<td>Number of key exchange payload messages received.</td>
</tr>
<tr>
<td>Total Number of Identity Payload Send</td>
<td>Number of ID payload messages sent.</td>
</tr>
<tr>
<td>Total Number of Identity Payload Rcv</td>
<td>Number of ID payload messages received.</td>
</tr>
<tr>
<td>Total Number of Signature Payload Send</td>
<td>Number of authentic payload messages sent.</td>
</tr>
<tr>
<td>Total Number of Signature Payload Rcv</td>
<td>Number of authentic payload messages received.</td>
</tr>
<tr>
<td>Total Number of Notify Payload Send</td>
<td>Number of notify payload messages sent.</td>
</tr>
<tr>
<td>Total Number of Notify Payload Rcv</td>
<td>Number of notify payload messages received.</td>
</tr>
<tr>
<td>Total Number of Delete Payload Send</td>
<td>Number of delete payload messages sent.</td>
</tr>
<tr>
<td>Total Number of Delete Payload Rcv</td>
<td>Number of delete payload messages received.</td>
</tr>
<tr>
<td>Total Number of Retransmissions</td>
<td>Number of messages retransmitted.</td>
</tr>
<tr>
<td>Total Number of Reestablishments Initiated</td>
<td>Number of phase 2 reestablishments initiated.</td>
</tr>
</tbody>
</table>
Sample Scenario

Scenario Description
This sample tests the normal behavior of ISAKMP implementation for a Tunnel in which same Security Policies (SP) is used for both inbound & outbound packets. And also the basic packet exchange during security association establishment.

Topology
Nodes 1 to 6 are connected by a wired point-to-point link as shown above. Node 3 and node 4 are negotiating at the interfaces specified in their respective configuration file. See Figure 13.

FIGURE 13. Topology of the ISAKMP Model

One CBR application is configured from node 1 to node 6.

Command Line Configuration
Include the following lines in the scenario configuration (.config) file:

```
# Nodes are connected through wired point to point link
#
# The LINK keyword is used to define a "point-to-point link".
#
LINK N8-192.168.1.0 { 1, 3 }
LINK N8-192.168.2.0 { 2, 3 }
LINK N8-192.168.3.0 { 3, 4 }
LINK N8-192.168.4.0 { 4, 5 }
LINK N8-192.168.5.0 { 4, 6 }
LINK N8-192.168.1.0 { 1, 3 }
LINK N8-192.168.2.0 { 2, 3 }
LINK N8-192.168.3.0 { 3, 4 }
LINK N8-192.168.4.0 { 4, 5 }
LINK N8-192.168.5.0 { 4, 6 }

[3] ISAKMP-SERVER YES

[4] ISAKMP-SERVER YES
```
ISAKMP Model

ISAKMP-PHASE-1-START-TIME 60
ISAKMP-ENABLE-IPSEC YES

Include the following lines in the file "node3.isakmp":

NODE 192.168.3.1
PEER 192.168.3.2 3-4-Config

3-4-Config
PHASE 1
DOI ISAKMP.DOI
EXCHANGE_TYPE EXCH_BASE
FLAGS ACE
TRANSFORMS 3DES-SHA

PHASE 2
LOCAL-ID-TYPE IPV4_ADDR_SUBNET
LOCAL-NETWORK 192.168.1.0
LOCAL-NETMASK 255.255.255.0
REMOTE-ID-TYPE IPV4_ADDR_SUBNET
REMOTE-NETWORK 192.168.4.0
REMOTE-NETMASK 255.255.255.0
UPPER-LAYER-PROTOCOL UDP
DOI IPSEC.DOI
EXCHANGE_TYPE EXCH_IDENT
FLAGS ACE
PROPOSALS ESP-DES-MD5-PFS AH-MD5-PFS
ESP-DES-MD5-PFS
PROTOCOLS ESP-DES-MD5-PFS
AH-MD5-PFS
PROTOCOLS AH-MD5-PFS
ESP-DES-MD5-PFS
PROTOCOL_ID ESP
TRANSFORMS ESP-DES-MD5-PFS-XF
AH-MD5-PFS
PROTOCOL_ID AH
TRANSFORMS AH-MD5-PFS-XF

Phase-1-Transforms

TRANSFORM_NAME 3DES-SHA
TRANSFORM_ID KEY_IKE
ENCRYPTION_ALGORITHM DEFAULT
HASH_ALGORITHM SHA
AUTHENTICATION_METHOD RSA_SIG
GROUP_DESCRIPTION MODP_1024
LIFE 10

TRANSFORM_NAME DES-MD5
TRANSFORM_ID KEY_IKE
ENCRYPTION_ALGORITHM DES-CBC
HASH_ALGORITHM MD5
AUTHENTICATION_METHOD PRE_SHARED
GROUP_DESCRIPTION MODP_1024
LIFE 7

Phase-2-Transforms

TRANSFORM_NAME ESP-DES-MD5-PFS-XF
TRANSFORM_ID ESP DES
ENCAPSULATION_MODE TUNNEL
GROUP_DESCRIPTION MODP_1024
AUTHENTICATION_ALGORITHM HMAC-MD5-96
LIFE 10

TRANSFORM_NAME AH-MD5-PFS-XF
TRANSFORM_ID AH_MD5
ENCAPSULATION_MODE TUNNEL
GROUP_DESCRIPTION MODP_1024
AUTHENTICATION_ALGORITHM HMAC-MD5
LIFE 15

Include the following lines in the file “node4.isakmp”:

    NODE 192.168.3.2
    PEER 192.168.3.1 4-3-Config

4-3-Config

PHASE 1

DOI ISAKMP DOI
EXCHANGE_TYPE EXCH_AGGR
FLAGS ACE
TRANSFORMS 3DES-SHA

PHASE 2

LOCAL-ID-TYPE IPV4_ADDR_SUBNET
LOCAL-NETWORK 192.168.4.0
LOCAL-NETMASK 255.255.255.0
REMOTE-ID-TYPE IPV4_ADDR_SUBNET
REMOTE-NETWORK 192.168.1.0
REMOTE-NETMASK 255.255.255.0
UPPER-LAYER-PROTOCOL UDP
DOI IPSEC_DOI
EXCHANGE_TYPE EXCH_IDENT
FLAGS ACE
PROPOSALS ESP-DES-MD5-PFS AH-MD5-PFS
FLAGS
ESP-DES-MD5-PFS
PROTOCOLS ESP-DES-MD5-PFS
AH-MD5-PFS
PROTOCOLS AH-MD5-PFS
ESP-DES-MD5-PFS
PROTOCOL_ID ESP
TRANSFORMS ESP-DES-MD5-PFS-XF
AH-MD5-PFS
PROTOCOL_ID AH
TRANSFORMS AH-MD5-PFS-XF
ISAKMP Model

Phase-1-Transforms

TRANSFORM_NAME 3DES-SHA
TRANSFORM_ID KEY_IKE
ENCRYPTION_ALGORITHM DEFAULT
HASH_ALGORITHM SHA
AUTHENTICATION_METHOD RSA_SIG
GROUP_DESCRIPTION MODP_1024
LIFE 5

TRANSFORM_NAME DES-MD5
TRANSFORM_ID KEY_IKE
ENCRYPTION_ALGORITHM DES-CBC
HASH_ALGORITHM MD5
AUTHENTICATION_METHOD PRE_SHARED
GROUP_DESCRIPTION MODP_1024
LIFE 7

Phase-2-Transforms

TRANSFORM_NAME ESP-DES-MD5-PFS-XF
TRANSFORM_ID ESP DES
ENCAPSULATION_MODE TUNNEL
GROUP_DESCRIPTION MODP_1024
AUTHENTICATION_ALGORITHM HMAC-MD5-96
LIFE 10

TRANSFORM_NAME AH-MD5-PFS-XF
TRANSFORM_ID AH_MD5
ENCAPSULATION_MODE TUNNEL
GROUP_DESCRIPTION MODP_1024
AUTHENTICATION_ALGORITHM HMAC-MD5
LIFE 15

GUI Configuration

To configure the sample scenario in QualNet GUI, perform the following steps:

1. Go to Hierarchy > Nodes > Host3 > Node configurations > Network Protocol > Enable ISAKMP?
   In the Configurable Property window, set Enable ISAKMP? to YES.

2. Create a *.isakmp file for each ISAKMP server as described earlier.

3. In the Configurable Property window, configure the ISAKMP parameters.
   a. Click ISAKMP Configuration File. When a Property Editor Window appears, specify the location of
      the ISAKMP configurable file for node 3 (*.isamkp file).

      Note: If default.isakmp file is specified, there must be an application between negotiating nodes.

   b. Configure ISAKMP-PHASE1-START-TIME to 3S.
   c. Set ISAKMP Enable IPSec? parameter to NO.

4. Similarly, configure Node4 to ISAKMP enabled.
Secure Neighbor Model

The secure neighbor authentication has two variants. The first variant is based on pair-wise shared secrets, and the second variant is based on certification.

In secure neighbor authentication (SNAuth), every mobile node establishes an authenticated neighborhood on the move. Periodically, every mobile node X broadcasts its identity packet <SNAuth-HELLO, X> to its neighborhood.

1. In the pair-wise shared secret variant of SNAuth, Y, a neighboring receiver of the identity broadcast initiates a 3-way challenge-response handshake to authenticate X, the sender of the identity broadcast.
   a. Suppose X and Y share a pair-wise secret k. Now Y selects a random nonce n1, encrypts n1 with k, sends the encrypted result ENCk(n1) to X by a message <CHALLENGE, Y, ENCk(n1)>.
   b. If the receiver of the challenge message is indeed X, then it can decrypt ENCk(n1) and sees n1. X selects another random nonce n2, encrypts ENCk(n1 XOR n2), and sends back <RESPONSE1, X, n2, ENCk(n1 XOR n2)> as the response to the challenger Y.
   c. When Y receives the response, Y decrypts ENCk(n1 XOR n2) and obtains n1 XOR n2. If Y can get the same result from XORing n2 in the response and its own challenge n1, then X passes the test with success. Otherwise, Y does not send any packet to X and does not receive packets from X except the response packets, until a correct <RESPONSE1> packet from X can pass the test. Upon detecting a success, Y puts X in its secure neighbor list. Y selects a random nonce n3 and sends out a confirmation response <RESPONSE2, Y, n3, ENCk(n1 XOR n2 XOR n3)> to X.
   d. Upon receiving the RESPONSE2 message, X decrypts ENCk(n1 XOR n2 XOR n3) and obtains n1 XOR n2 XOR n3. If this matches the result of XORing n1 that is previously decrypted, its own n2 and n3 in the RESPONSE2 packet, then X inserts Y into its secure neighbor list. (This three-way handshake is required because X needs to verify that Y actually knows k)
   e. End of the challenge-response protocol.

* The cryptographic term, “nonce” is used above to mean a value that is used only once.
In the above description, all nonce length is currently set to 128-bit long. Encryption block length is 128-bit. Key k can be 128-bit, 192-bit, or 256-bit. Session key means that the key n1 is used until the time when the next HELLO received by Y from X successfully passes the test again.

2. A slightly different challenge-response scheme is used if Y does not pre-share a master secret k with X. Here X must broadcast its certificate CERT\_X = [X, certified public key PK\_X, certificate valid time] in a CERTIFIED\_HELLO message. For Y’s CHALLENGE, Y uses PK\_X to encrypt n1 and obtains ciphertext PK\_X(n1). Y must also add its own certificate CERT\_Y = [Y, certified public key PK\_Y, certificate valid time] and sign the entire message with its own private key SK\_Y. We recommend the public key cryptosystem in use be an Elliptic Curve Cryptosystem (ECC), because ECC features shorter certificate length and ciphertext length, thus incurring less communication overhead.

As depicted below, there are a number of computational changes, and RESPONSE2 is spared, but the RESPONSE message format is unchanged.

When every neighboring receiver of X finishes the authentication and key-agreement process, node X obtains a secure snapshot of its neighborhood. In the neighborhood, every other node is authenticated and
shares an IPsec security association with the node X. As the SNAuth protocol runs on every mobile node, the statement is true if node X is replaced with any node X'.

Caveats

- All the above secure neighbor authentication variants may fail to reach the session key establishment final phase due to jamming, packet loss, etc. In other words, the adversary can deny the protocol execution. However, the adversary cannot forge (uncompromised) neighboring nodes' identities.
- Brute-force jamming and wormhole attacks are feasible attacks to foil secure neighbor authentication. Brute-force jamming can be thwarted by countermeasures such as spread spectrum and forward error correction. Wormhole attack can be thwarted by countermeasures such as distance-bounding protocols. These attacks are not studied here.

SNAuth is a building block for other advanced network security services. For example, in secure routing, you can enforce a rule that the current node only forwards packets for those nodes detected by SNAuth. Packets from other nodes not detected by SNAuth are dropped. This way, packets from unauthenticated nodes are limited in their immediate neighborhoods. The danger of denial-of-service is hence limited in unauthenticated nodes' immediate neighborhoods.

Command Line Configuration

To enable secure neighbor, include the following parameter in the scenario configuration (.config) file:

```
[<node-ID>] SECURE-NEIGHBOR-ENABLED YES
```

The default value of this parameter is NO.

The secure neighbor configuration parameters are described in Table 10.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECURE-NEIGHBOR-TIMEOUT</td>
<td>&lt;timeout&gt;</td>
</tr>
<tr>
<td></td>
<td>This parameter controls the period for announcing certified credentials.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For fast mobile scenarios, reduce the value to get fresher snapshots. For slow mobile scenarios, enlarge the value to reduce overhead.</td>
</tr>
<tr>
<td></td>
<td>This is an optional parameter.</td>
</tr>
<tr>
<td></td>
<td>The default value is 5S.</td>
</tr>
</tbody>
</table>
GUI Configuration

To configure secure neighbor in the GUI, perform the following steps:

2. In the Configurable Property window, set Enable Secure-neighborhood? to YES, as shown in the Figure 14.
3. In the Configurable Property window, configure the secure neighbor parameters, as shown in the Figure 15.
FIGURE 15. Configuring Secure Neighbor Parameters

Statistics
Table 11 shows the statistics collected by Secure Neighbor.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of HELLO Initiated</td>
<td>Total number of rounds of Hello messages sent.</td>
</tr>
<tr>
<td>Number of HELLO Received</td>
<td>Total number of Hello messages received.</td>
</tr>
<tr>
<td>Number of CHALLENGE Initiated</td>
<td>Total number of Challenge messages sent.</td>
</tr>
<tr>
<td>Number of CHALLENGE Received</td>
<td>Total number of Challenge messages received.</td>
</tr>
<tr>
<td>Number of RESPONSE1 Initiated</td>
<td>Total number of Response1 messages sent.</td>
</tr>
<tr>
<td>Number of RESPONSE1 Received</td>
<td>Total number of Response1 messages received.</td>
</tr>
<tr>
<td>Number of RESPONSE2 Initiated</td>
<td>Total number of Response2 messages sent.</td>
</tr>
<tr>
<td>Number of RESPONSE2 Received</td>
<td>Total number of Response2 messages received.</td>
</tr>
</tbody>
</table>

Sample Scenario

Scenario Description
This sample scenario tests the secure neighbor implementation in a simple wireless scenario. Nodes 1 and 2 are connected through a wireless subnet. Both the nodes are secure neighbor enabled. See Figure 16.
FIGURE 16. Secure Neighbor Sample Scenario Topology

Command Line Configuration
To configure the sample scenario, include the following lines in the scenario configuration (.config) file:

```
SECURE-NEIGHBOR-ENABLED YES
SECURE-NEIGHBOR-TIMEOUT 5S
SECURE-NEIGHBOR-CERTIFIED-HELLO NO
```

GUI Configuration
To configure the sample scenario in QualNet GUI, perform the following steps:

1. Create a new scenario using the Scenario Designer.
2. Go to ConfigSettings > Network Protocols > Network Protocol > Enable Secure-neighborhood?.
3. In the Configurable Property window, set Enable Secure-neighborhood? to YES.
4. In the Configurable Property window, configure the secure neighbor configurable parameters, as shown in Figure 17.
   a. Set Secure-neighborhood expiration timeout to the default value of 5 seconds.
   b. Set Secure-neighborhood certified hello? to its default value of NO.
FIGURE 17. Configuring Secure Neighborhood Parameters for Sample Scenario
WEP and CCMP Models

Wired Equivalent Privacy (WEP)

Wired Equivalent Privacy (WEP) is a MAC layer security protocol that provides security for wireless LANs equivalent to security provided in wired LANs.

In WEP, a secret key is distributed to cooperating STAs using an external key management path, independent of the MAC layer. The secret key combined with an Initialization Vector (IV) resulting in a seed is given as an input to a Pseudo-Random Number Generator (PRNG). The PRNG outputs a key sequence (k) of pseudorandom octets.

An integrity algorithm operates on plaintext data to produce an ICV to protect against unauthorized data modification. The key sequence (k) is combined with the plaintext concatenated with the ICV to generate the cipher text. The secret key remains constant while the IV changes periodically. Thus, there is a one-to-one correspondence between the IV and k.

The WEP algorithm is applied to the frame body of an MPDU. The (IV, frame body, ICV) triplet forms the actual data to be sent in the data frame.

CTR with CBC-MAC Protocol (CCMP)

CCMP (CTR with CBC-MAC Protocol) is an RSNA data confidentiality and integrity protocol.

WEP is known to be insecure and is replaced by CCMP. CCMP is based on the CCM of the AES encryption algorithm. CCM is a generic authenticate-and-encrypt block cipher mode. A unique temporal key (for each session) and a unique nonce value (a value that's used only once for each frame) are required for protecting the MPDUs. CCMP uses a 48-bit Packet Number (PN) to protect the MPDUs.

Note: The PN is never repeated for a series of encrypted MPDUs using the same temporal key.

CCMP encrypts the payload of a plaintext MPDU and encapsulates the resulting cipher text using the following:

1. Increment the PN, so that each MPDU has a unique PN for the same temporal key.
2. Use the fields in the MPDU header to construct the additional authentication data (AAD) for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD.
3. Construct the CCM Nonce block from the PN, A2, and the Priority field of the MPDU where A2 is MPDU Address 2. The Priority field has a reserved value set to 0.
4. Place the new PN and the key identifier into the 8-octet CCMP header.
5. Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and MIC. This step is known as CCMP originator processing.
6. Form the encrypted MPDU by combining the original MPDU header, the CCMP header, the encrypted data and MIC, as described in IEEE 802.11i-2004 Standard, Sec-8.3.3.2.

CCMP decrypts the payload of a cipher text MPDU and decapsulates plaintext MPDU using the following:

1. The encrypted MPDU is parsed to construct the AAD and nonce values.
2. The AAD is formed from the MPDU header of the encrypted MPDU.
3. The nonce value is constructed from the A2, PN, and Priority Octet fields (reserved and set to 0).
4. The MIC is extracted for use in the CCM integrity checking.
5. The CCM recipient processing uses the temporal key, AAD, nonce, MIC, and MPDU cipher text data to recover the MPDU plaintext data and, to check the integrity of the AAD and MPDU plaintext data.
6. The received MPDU header and the MPDU plaintext data from the CCM recipient processing can be concatenated to form a plaintext MPDU.
7. The decryption processing prevents replay of MPDUs by validating that the PN in the MPDU is greater than the replay counter maintained for the session.

The decapsulation process succeeds when the calculated MIC matches the MIC value obtained from decrypting the received encrypted MPDU. The original MPDU header is concatenated with the plaintext data resulting from the successful CCM recipient processing to create the plaintext MPDU.

**Command Line Configuration**

To enable WEP or CCMP, include the following parameter in the scenario configuration (.config) file:

```
MAC-SECURITY-PROTOCOL [WEP | CCMP]
```

*Note*: In order to run WEP or CCMP, the MAC protocol must be configured to be 802.11 MAC. See the 802.1 MAC protocol section of QualNet 4.5 Wireless Model Library for details.

The WEP/CCMP configuration parameters are described in Table 10.

**TABLE 12. WEP/CCMP Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEP-RC4-DELAY</td>
<td>This parameter specifies the crypto delay for WEP.</td>
</tr>
<tr>
<td></td>
<td>This is an optional parameter.</td>
</tr>
<tr>
<td></td>
<td>The default value is 10US.</td>
</tr>
<tr>
<td>CCMP-AES-DELAY</td>
<td>This parameter specifies the crypto delay for CCMP.</td>
</tr>
<tr>
<td></td>
<td>This is an optional parameter.</td>
</tr>
<tr>
<td></td>
<td>The default value is 10US.</td>
</tr>
<tr>
<td>WEP-CCMP-ALLOW-UNENC</td>
<td>This parameter enables the MAC service interface to accept received MPDUs</td>
</tr>
<tr>
<td></td>
<td>which are unencrypted.</td>
</tr>
<tr>
<td></td>
<td>This is an optional parameter.</td>
</tr>
<tr>
<td></td>
<td>The default value is NO.</td>
</tr>
</tbody>
</table>
WEP and CCMP Models

Format of the WEP and CCMP Configuration Files

The WEP configuration file and the CCMP configuration file have the same format.

You need to specify the KeyMappings table configuration. There is a one-one key mappings table defined per destination (RA) for a given node as:

```
KeyMappings <TA> <RA> <Key Type> <Key>
```

where:

- **TA** is the transmitter address. It can be `<node_id | interface address | subnet address>.
- **RA** is the receiver address. It can be `<interface address | subnet address>.
- **Key Type** is a string. For .wep files the only possible value is WEP while for .ccmp it can be WEP or CCMP.
- **Key** is the actual key value used for encryption/decryption. It will be a string.

For example, if you have two nodes (node 1 and 2). The entries in the file will be as follows:

```
KeyMappings 1 192.168.0.2 WEP ffa0
KeyMappings 192.168.0.1 192.168.0.2 CCMP ffa0
```

GUI Configuration

To configure WEP or CCMP in QualNet GUI, perform the following steps:

1. Go to **Hierarchy > Nodes > Wireless Subnet [Network address] > Wireless Subnet Properties > MAC Protocol**. In the Configurable Property window, set **MAC Protocol** to **802.11**, as shown in Figure 18.
2. Go to Hierarchy > Nodes > Wireless Subnet [Network address] > Wireless Subnet Properties > MAC Protocol > Security Protocol. In the Configurable Property window, set MAC Security Protocol to WEP or CCMP and configure the other parameters for the selected protocol, as shown in Figure 19.
Statistics
The WEP/CCMP statistics are shown in Table 13.

**TABLE 13. WEP/CCMP Statistics**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WEP</strong></td>
<td></td>
</tr>
<tr>
<td>MAC, WEP Packets Encrypted</td>
<td>Number of WEP encrypted packets.</td>
</tr>
<tr>
<td>MAC, WEP Packets Decrypted</td>
<td>Number of WEP decrypted packets.</td>
</tr>
<tr>
<td>MAC, WEP Packets Discarded</td>
<td>Number of non-WEP packets discarded by a STA on reception.</td>
</tr>
<tr>
<td>MAC, WEP Packets Undecrypted</td>
<td>Number of protected packets unable to decrypt.</td>
</tr>
<tr>
<td><strong>CCMP</strong></td>
<td></td>
</tr>
<tr>
<td>MAC, CCMP Packets Encrypted</td>
<td>Number of CCMP encrypted packets.</td>
</tr>
<tr>
<td>MAC, CCMP Packets Decrypted</td>
<td>Number of CCMP decrypted packets.</td>
</tr>
<tr>
<td>MAC, CCMP Packets Discarded</td>
<td>Number of non-WEP packets discarded by a STA on reception.</td>
</tr>
<tr>
<td>MAC, CCMP Packets Undecrypted</td>
<td>Number of protected packets unable to decrypt.</td>
</tr>
</tbody>
</table>

FIGURE 19. Configuring WEP as the Security Protocol
Sample Scenario

Scenario Description
In the sample scenario, five nodes (nodes 1 through 5) are connected through a wireless subnet. WEP or CCMP is enabled for the subnet.

Command Line Configuration

WEP scenario
To configure the sample scenario using WEP, include the following lines in the scenario configuration (.config) file:

```
SUBNET N8-192.0.0.0 { 1 thru 5 } 451.95 1145.77 0.0
[ N8-192.0.0.0 ] MAC-PROTOCOL MACDOT11
[ N8-192.0.0.0 ] MAC-SECURITY-PROTOCOL WEP
[ N8-192.0.0.0 ] WEP-RC4-DELAY 5US
[ N8-192.0.0.0 ] WEP-CCMP-ALLOW-UNENC YES
[ N8-192.0.0.0 ] WEP-CONFIG-FILE wirelesssubnet-wep-on.wep
[ N8-192.0.0.0 ] NETWORK-PROTOCOL IP
```

Include the following lines int he WEP configuration file “wirelesssubnet-wep-on.wep”:

```
KeyMappings 1 192.0.0.3 WEP ffa0
KeyMappings 192.0.0.3 192.0.0.1 WEP ffa0
```

CCMP scenario
To configure the sample scenario using CCMP, include the following lines in the scenario configuration (.config) file:

```
SUBNET N8-192.0.0.0 { 1 thru 5 } 451.95 1145.77 0.0
[ N8-192.0.0.0 ] MAC-PROTOCOL MACDOT11
[ N8-192.0.0.0 ] MAC-SECURITY-PROTOCOL CCMP
[ N8-192.0.0.0 ] CCMP-AES-DELAY 5US
[ N8-192.0.0.0 ] CCMP-CONFIG-FILE wirelesssubnet-ccmp-on.ccmp
```

Include the following lines int he WEP configuration file “wirelesssubnet-ccmp-on.ccmp”:

```
KeyMappings 1 192.0.0.3 CCMP ffa0
KeyMappings 192.0.0.3 192.0.0.1 CCMP ffa0
```