**Thách thức về an ninh trong mạng MANET**

- Môi trường truyền sóng điện từ là không khí, vì vậy nguy cơ bị nghe

trộm là rất lớn, từ đó kẻ tấn công có thể phân tích lưu lượng mạng phục vụ cho

các mục đích tấn công tiếp theo.

- Việc các nút gia nhập và rời mạng bất kỳ lúc nào tạo nên sự thay đổi

thường xuyên về cấu trúc mạng đòi hỏi các giao thức định tuyến liên tục phát

các yêu cầu quảng bá trong toàn mạng cũng dẫn đến việc mất an ninh trong

mạng. Đồng thời, việc cấu trúc mạng liên tục thay đổi cũng là một khó khăn để

các giao thức định tuyến phát hiện rằng thông điệp điều khiển sai lệch được sinh

ra bởi nút độc hại hay là do quá trình thay đổi cấu trúc mạng.

- Giới hạn về tài nguyên như băng thông và năng lượng làm giảm khả năng chống đỡ của mạng trước các cuộc tấn công.

- Thiếu một cơ sở hạ tầng trợ giúp gây khó khăn khi triển khai các cơ chế bảo mật trong mạng.

**2.1.2. Các yêu cầu về an ninh**

Để đảm bảo an toàn trong giao thức định tuyến mạng MANET

- **Tính bảo mật (Confidentiality):** Đảm bảo thông điệp truyền trong mạng

phải được giữ bí mật. Trong một số trường hợp cần đảm bảo bí mật cả với các

thông điệp định tuyến quảng bá trong mạng vì từ thông tin các thông điệp này có

thể khai thác một số thông tin giúp ích cho việc tấn công.

- **Tính xác thực (Authentication):** Đảm bảo một nút phải xác định được

danh tính rõ ràng của một nút khác trong quá trình truyền dữ liệu với nó.

- **Tính toàn vẹn (Integrity):** Đảm bảo các thông điệp không bị chỉnh sửa

trong toàn bộ quá trình truyền.

- **Tính chống chối bỏ (Non-Repudiation):** Đảm bảo luôn xác định được

nguồn gốc thông điệp truyền từ nút nào.

- **Tính sẵn sàng (Availability):** Đảm bảo tính sẵn sàng của các nút mặc dù

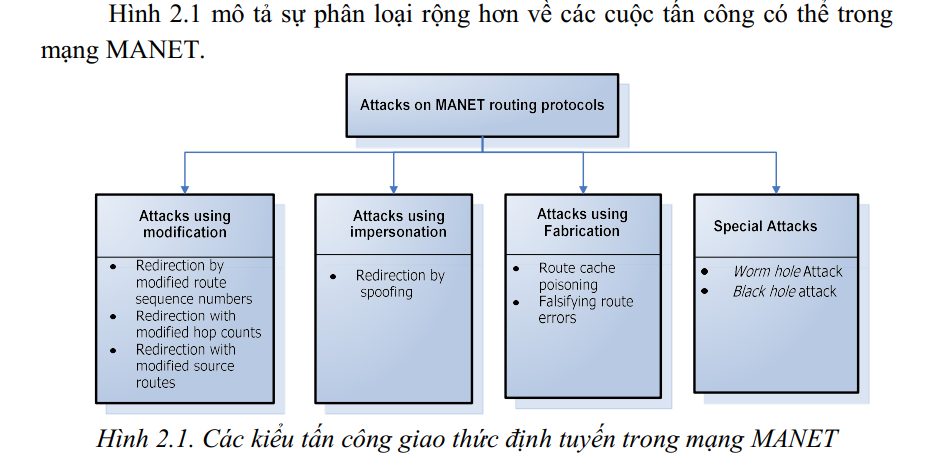
bị các cuộc tấn công. Trong đó tấn công từ chối dịch vụ đe dọa tới bất kỳ tầng

nào trong mạng ad hoc. Ở tầng điều khiển môi trường truy nhập, kẻ tấn công có thể sử dụng hình thức chèn ép kênh truyền vật lý; ở tầng mạng sự gián đoạn

trong hoạt động của các giao thức định tuyến, ở các tầng cao hơn có thể là tấn

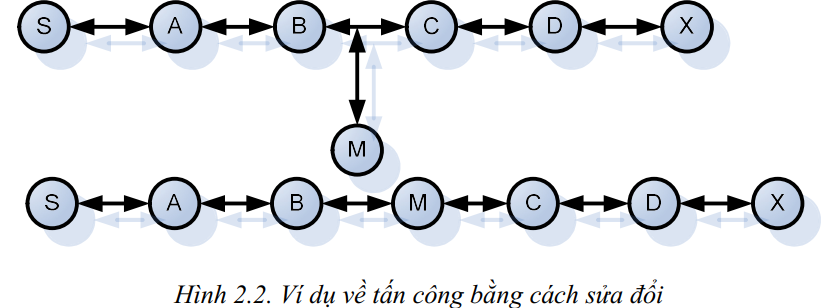
công vào các ứng dụng bảo mật ví dụ như hệ thống quản lý khóa.

**Các phương thức tấn công trong giao thức định tuyến mạng MANET**

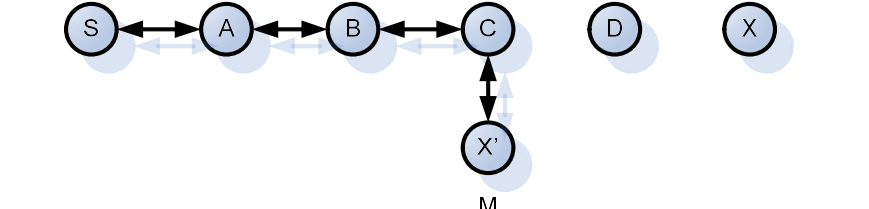


- Sửa đổi số tuần tự đích (destination sequence number), số chặng(hop\_count) của tuyến đường: Thể hiện rõ ràng nhất trong giao thức định tuyến  
AODV. Kẻ tấn công sửa đổi số tuần tự đích, hoặc số chặng của gói tin yêu cầu  
tuyến (RREQ) hoặc gói tin trả lời tuyến (RREP) để tạo nên tuyến đường mới có  
hiệu lực, qua đó chiếm quyền điều khiển quá trình truyền dữ liệu từ nguồn tới  
đích.

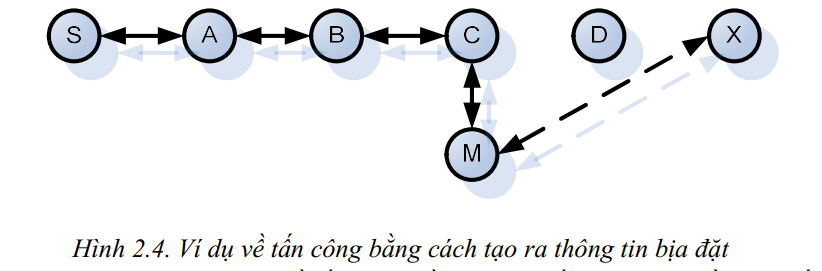
- *Sửa đổi nguồn của tuyến đường*: Thể hiện rõ trong giao thức định tuyến  
nguồn DSR. Kẻ tấn công chặn thông điệp sửa đổi danh sách các nguồn trước khi  
gửi tới nút đích trong quá trình truyền



***2.2.2. Tấn công bằng cách mạo danh***Kiểu tấn công bằng cách mạo danh đe dọa tính xác thực và bảo mật  
trong mạng. Nút độc hại có thể giả mạo địa chỉ của nút khác để thay đổi cấu trúc  
mạng hoặc ẩn mình trên mạng.  
Nút độc hại mạo danh bằng cách thay đổi địa chỉ IP nguồn trong thông  
điệp điều khiển. Một lý do khác để mạo danh là để thuyết phục các nút khác  
thay đổi thông tin trong bảng định tuyến của chúng rằng tôi là một nút tin cậy,  
cách này còn được biết đến như tấn công vào bảng định tuyến.  
Một trong những điển hình của loại tấn công này là “Man in the midle  
attack”. Nút độc hại thực hiện cuộc tấn công này bằng cách kết hợp giữa mạo  
danh và loại bỏ gói tin. Về mô hình vật lý, nó phải được chọn sao cho là nút nằm  
trong phạm vi tới đích, tức là nằm giữa tuyến đường để có thể chặn bất kỳ thông  
tin nào từ nút khác tới đích. Để thực hiện việc này, khi nút nguồn gửi RREQ, nút  
độc hại loại bỏ RREQ và gửi trả lời RREP giả mạo như là nút đích trả lời, song  
song với quá trình này nó gửi gói tin RREQ tới nút đích và loại bỏ gói RREP từ  
nút đích trả lời. Bằng cách này, nút độc hại đứng giữa nắm giữ thông tin trong  
quá trình truyền thông nút nguồn, nút đích. Với thông tin thu được, sử dụng các ộc hại loại bỏ RREQ và gửi trả lời RREP giả mạo như là nút đích trả lời, song  
ng với quá trình này nó gửi gói tin RREQ tới nút đích và loại bỏ gói RREP từ  
nút đích trả lời. Bằng cách này, nút độc hại đứng giữa nắm giữ thông tin trong  
quá trình truyền thông nút nguồn, nút đích. Với thông tin thu được, sử dụng các  
phương pháp bảo mật ở các lớp trên, kẻ tấn công có thể giải mã hoặc tìm ra các  
khóa để tiếp tục quá trình tấn công vào các lớp phía trên của hệ thống

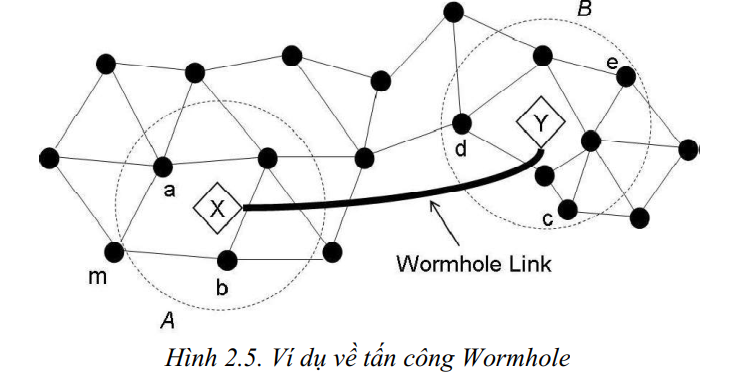


***2.2.3. Tấn công bằng cách tạo ra thông tin bịa đặt***



Trong cách tấn công này [18], nút độc hại cố gắng để “bơm” vào mạng  
các thông điệp giả mạo hoặc các thông điệp định tuyến sai để phá vỡ cơ chế  
định tuyến trong mạng

***2.3.4. Một vài kiểu tấn công đặc biệt  
- Tấn công lỗ sâu (Wormhole Attack):***Trong một cuộc tấn công kiểu  
wormhole , hai hoặc nhiều hơn các nút độc hại thông đồng với nhau bằng  
cách thiết lập một đường. Trong giai đoạn tìm đường của giao thức định tuyến,  
thông báo yêu cầu định tuyến được chuyển tiếp giữa các nút độc hại sử dụng các  
đường hầm đã được thiết lập sẵn. Do đó, thông báo yêu cầu định tuyến đầu tiên  
đến đích là một trong những thông báo được chuyển tiếp từ nút độc hại. Do vậy  
nút độc hại được thêm vào trong đường dẫn từ nút nguồn đến nút đích. Một khi  
các nút độc hại đã có trong đường dẫn định tuyến, các nút độc hại hoặc sẽ bỏ tất  
cả các gói dữ liệu dẫn đến việc từ chối hoàn toàn dịch vụ, hoặc loại bỏ có chọn  
lọc gói tin để tránh bị phát hiện.



Ở hình 2.5 là một ví dụ về tấn công wormhole. Nút X và nút Y là hai đầu  
của kiểu tấn công wormhole. Nút X phát lại mọi gói tin trong vùng A mà nút Y  
nhận được trong vùng B và ngược lại. Dẫn tới việc tất cả các nút trong vùng A  
cho rằng là hàng xóm của các nút trong vùng B và ngược lại. Kết quả là, nút X  
và nút Y dễ dàng tham gia vào tuyến đường truyền dữ liệu. Khi đó chúng chỉ  
việc hủy bỏ mọi gói tin truyền qua chúng và đánh sập mạng.  
*-* ***Tấn công lỗ đen (Blackhole Attack):***Trong cuộc tấn công lỗ đen [10],  
nút độc hại tuyên bố rằng nó có tuyến đường hợp lệ tới tất cả các nút khác trong  
mạng để chiếm quyền điều khiển lưu lượng giữa các thực thể truyền. Sau khi  
nhận dữ liệu truyền, nó không chuyển tiếp mà loại bỏ (drop) tất cả các gói tin  
này. Do đó, nút lỗ đen này có thể giám sát và phân tích lưu lượng toàn bộ các  
nút trong mạng mà nó đã tấn công.

***\* Quy trình mô phỏng mạng MANET với NS-2:***Quá trình mô phỏng mạng MANET với bộ mô phỏng NS-2 [1] thông  
thường trải qua các bước như sau trong việc xây dựng tệp kịch bản:  
- Tạo bộ lập lịch các sự kiện  
+ Tạo bộ lập lịch  
*set ns\_ [new Simulator]*+ Lập lịch sự kiện  
*$ns at <time><event>*

+ Khởi động bộ lập lịch  
*$ns run*- Ghi lại vết các sự kiện của mạng mô phỏng  
+ Dò vết các gói tin trên tất cả các liên kết, xuất ra tệp traceout.tr  
*$ns trace-all [open traceout.tr w]*+ Dò vết các gói tin trên tất cả các liên kết, xuất ra định dạng dùng t  
*$ns namtrace-all [open traceout.nam w]*+ Cũng có thể dò vết một số tham số, ví dụ:  
*$ns trace-queue $n0 $n1  
$ns nam trace-queue $n0 $n1*- Thiết lập mạng mô phỏng  
+ Thiết lập tô-pô mạng  
*set topo [new Topography]  
$topo load\_flatgrid $val(x) $val(y)  
# Create God  
create-god $val(nn)*- Cấu hình nút di động  
+ Cấu hình nút di động  
*$ns\_ node-config -adhocRouting $val(rp) \  
-llType $val(ll) \  
-macType $val(mac) \  
-ifqType $val(ifq) \  
-ifqLen $val(ifqlen) \  
-antType $val(ant) \  
-propType $val(prop) \  
-phyType $val(netif) \  
-topoInstance $topo \  
-agentTrace ON \  
-routerTrace ON \  
-macTrace ON \  
-movementTrace ON \  
-channel $chan\_1\_*+ Thiết lập vị trí ban đầu  
*$node\_(0) set X\_ 1.0  
$node\_(0) set Y\_ 3.0  
$node\_(0) set Z\_ 0.0*

$node\_(1) set X\_ 4.0$node\_(1) set Y\_ 6.0$node\_(1) set Z\_ 0.0+ Thiết lập sự di chuyển cho các nút di động ta dùng  
$ns\_ at 10.0 “$node\_(0) setdest 20.0 10.0 15.0”- Tạo ra các nguồn sinh lưu lượng  
+ Tạo lưu lượng  
set udp\_(0) [new Agent/UDP]$ns\_ attach-agent $node\_(0) $udp\_(0)set null\_(0) [new Agent/Null]$ns\_ attach-agent $node\_(1) $null\_(0)set cbr\_(0) [new Application/Traffic/CBR]$cbr\_(0) set packetSize\_ 512$cbr\_(0) set interval\_ 4.0$cbr\_(0) set random\_ 1$cbr\_(0) set maxpkts\_ 10000$cbr\_(0) attach-agent $udp\_(0)$ns\_ connect $udp\_(0) $null\_(0)$ns\_ at 158.7047093914378 "$cbr\_(0) start"+ Việc tạo lưu lượng có thể lấy từ tệp sinh lưu lượng của công cụ có sẵn  
cbrgen trong bộ mô phỏng NS-2.