Secure AODV in MANET

Software Recommended: NetSim Standard v11.1 (32/64-bit), Microsoft Visual Studio 2015/2017

Follow the instructions specified in the following link to clone/download the project folder from GitHub using Visual Studio:

https://tetcos.freshdesk.com/support/solutions/articles/14000099351-how-to-clone-netsim-file-exchange-project-repositories-from-github-

Other tools such as GitHub Desktop, SVN Client, Sourcetree, Git from the command line, or any client you like to clone the Git repository.

Note: It is recommended not to download the project as an archive (compressed zip) to avoid incompatibility while importing workspaces into NetSim.

Secure URL for the GitHub repository:

https://github.com/NetSim-TETCOS/Secure_AODV_Project_v11.1.git

Introduction:

SAODV is an extension of the AODV routing protocol that can be used to protect the route discovery mechanism providing security features like integrity and authentication. The reason only route discovery is secured by AODV is because data messages can be protected using a point-to-point security protocol like IPSec. SAODV uses a key management system and each node maintains public keys, encryption keys and decryption keys.

To implement SAODV, we have added **Secure AODV.c**, **RSA.c** and **Malicious.c** files in AODV project. RSA.c file is used to generate keys, encrypt and decrypt the data. Users can implement their own encryption algorithms by changing RSA.c file. Malicious.c file is used to identify malicious nodes present in the network.

Steps:

1. The downloaded project folder contains the folders Documentation and Secure_AODV_Workspace directory as shown below:

	<mark></mark>	ecure_A	ODV_Pr	roject_V11.1			-		×
File	Home	Share	e Vi	ïew					^ ?
Pin to Qui access	ck Copy	D Paste	א יייי ו	🖕 Move to ▾	X Delete ▼ ■I Rename	New folder	Properties	Select	
	Clipboard	d		Orgar	nize	New	Open		
← → × ↑ 📑 > File > Secure_AODV_Project v ♂ Search Secure_AODV_Project ル						,o			
		^	Nar	me	^	~	Date modified	Тур	e
ې کې	iick access ieDrive ittachment	ts		Documentaior Secure_AODV_	n _Workspace		27-03-2019 15:25 27-03-2019 15:28	5 File 8 File	folder folder
	ocuments	\checkmark	<						>
2 items	2 items 🔠 📰								

2. Import Secure_AODV_Workspace by going to Open Simulation->Workspace Options->More Options in NetSim Home window. Then select Import as shown below:

NetSim Home						×
NetSim Sta Network Simulation/Em Version 11.1.11 (64 Bit)	indard ^{ulation Platfo}	rm			www.tetcos.co	m
		Current workspace: NetSim_1	1.1.11_64_std_default			
New Simulation	Ctrl+N	Workspace name	Location	Description		
Onen Simulation	CHI-O	NetSim_11.1.11_64_std_default	C:\Users\TETCOS-PC	NetSim default Workspace	Export 🗊	
Examples License Settings		New	port Set as Current		Back	
Support		Learn		Documentation	Contact us	
Answers/FAQ Contact Technical Support Email - support@tetcos.	ort .com	Videos Experiments Manu	al T S	Jser Manual echnology Libraries iource Code Help	Email - sales@tetcos.com Phone - +91 767 605 4321	

3. It displays a window where users need to give the path of the workspace folder and click on OK as shown below:

M Import Workspace						
Analyse the content of your folder or archive file to find projects and import them. Browse the respective Workspace folder and import it as a current working Workspace. This will import all the folders experiments.						
Import from	Select path to import experiment from					
	OK Cancel					

4. Browse to the Secure_AODV_Workspace folder and click on select folder as shown below:

Select Folder							×
← → • ↑	« Se	ecure_A >	Secure_AODV_Workspace	~ Ū	Search Secure_	AODV_Worksp	, p
Organize 🔻 Ne	ew fold	ler				•==- ▼	?
a OneDrive	^	Name	^	Da	ate modified	Туре	
Attachments		bin		22	2-03-2019 09:11	File folder	
Documents		lcor	15	22	2-03-2019 09:12	File folder	
Pictures		Sec	ure_AODV_Example	22	2-03-2019 09:12	File folder	
This PC This PC 3D Objects Desktop Comments Documents Downloads Music Pictures		src .		22	2-03-2019 09:12	File folder	
📑 Videos	Folde	< er: Secure_	AODV_Workspace	[Select Folder	Cancel	>

- 5. After this click on OK button in the Import Workspace window.
- **6.** While importing the workspace, if the following warning message indicating Software Version Mismatch is displayed, you can ignore it and proceed.



7. The Imported workspace will be set as the current workspace automatically. To see the imported workspace, click on Open Simulation->Workspace Options->More Options as shown below:

💦 NetSim Home						- 🗆 X
NetSim St Network Simulation/I Version 11.1.11 (64 Bi	tandard Emulation Platfc it)	orm				www.tetcos.com
	Current workspace: Secure_AODV_Workspace					
New Simulation	Ctrl+N	Workspace name	Location 🔻	Description		
L On an Simulation	Chillio O	Secure_AODV_Workspace	C:\Users\NetSim\Desktop\File_Exch	3	Export	Ē
Open Simulation	Ctri+O	NetSim_11.1.11_64_std_default	C:\Users\NetSim	NetSim default Workspace	Export	Ē
License Settings		New	port Set as Current			Back
Support		Learn	D	ocumentation	Contact us	
Answers/FAQ Contact Technical Su Email - support@tetc	pport :os.com	Videos Experiments Manu	al Te So	er Manual Ihnology Libraries urce Code Help	Email - sales@tet Phone - +91 767	cos.com 605 4321

8. Open the Source codes in Visual Studio by going to Open Simulation-> Workspace Options and Clicking on Open code button as shown below:

NetSim Home						—	
NetSim Standard Network Simulation/Emulation Platfor Version 11.1.11 (64 Bit)	l					www.	tetcos.com
	Current workspace: Secure_AODV_Wor	kspace		Q Experime	nt name		
New Simulation Ctrl+N	Experiment name	Date modified	Network type				
Open Simulation Ctrl+O	Secure_AODV_Example	27-03-2019	MANET		View Results	Export	1
Examples License Settings	Open code Reset Code	: Reset Binari	es More optic	ons		Back	
Support	Learn		Documentation		Contact u	JS	
Answers/FAQ Contact Technical Support Email - support@tetcos.com	Videos Experiments Manual		User Manual Technology Libraries Source Code Help		Email - sa Phone - +	les@tetcos.con 91 767 605 43	n 21

9. Under the AODV project in the solution explorer you will be able to see that Malicious.c and Secure_AODV.c files which contain source codes which implements SAODV in NetSim respectively.

10. Based on whether you are using NetSim 32 bit or 64 bit setup you can configure Visual studio to build 32 bit or 64 bit Dll files respectively as shown below:

A	🕽 NetSim - Microsoft Visual Studio 💦 🕇 🖉	Quick Launch (Ctrl+Q)
File	e Edit View Project Build Debug Team Tools Test Analyze Window Help	ThoMaS SaNthosH 👻 TS
8	🖸 - 🔘 🔯 - 😭 🔛 🔐 🦻 - 🔆 - Debug - Vin32 - 🕨 Local Windows Debugger - 👼 🖕	
Servi	Win32 x64	Solution Explorer 🔹 🖣 🗙
er Exp	Configuration Manager	○ ○ 🏠 🖆 - │ `o - ≒ 🗗 🐚 🔎 💻
plorer		Search Solution Explorer (Ctrl+;)
5		g. Solution 'NetSim' (1 project)
olbo		A Let References
×		External Dependencies
		C AODV.c
		C AODV CheckRouteFound.c
		C AODV_RouteError.c
		C FIFOBuffer.c
		P C GeneralPacketProcessing.c b C HelloMersage c
		 C Malicious.c
		▷ C RouteMaint.c
		▷ C RouteTable.c
		C RREP.c
		b c RSA.c
		C Secure_AODV.c
	Output v li x	1
	Show output from: • 운 함 홈 점 점	
	Franci jet Output	
	Restly	Add to Source Control

11. Right click on the AODV project in the solution explorer and select Rebuild.

1 NetSim - Microsoft Visual Studio			T	🔐 🛛 Quick Laur	nch (Ctrl+Q) 🔑 🗕 🗗 🗙
File Edit View Project Build Debug Team Tools Test Analyze Window He	elp				ThoMaS SaNthosH 👻 TS
	Debugger 👻 👼 💷				_
Server Explorer		.*.	Roild	Solution Search S	Explorer
oot a state of the			Rebuild		AODV
u ga		□ □ ○ ○ ○	Clean View Analyze Project Only Retarget Projects Scope to This New Solution Explorer View Build Dependencies Add Class Wizard Manage NuGet Packages	, , , Ctrl+Shift+X	
Output			Debug	•	
Show output from: CMake	3 ³ 2		Source Control	+	
		ж	Cut	Ctrl+X	
			Paste	Ctrl+V	
		×	Remove	Del	
		<u>[x</u>	Rename		
			Unload Project		
▲			Rescan Solution		
Error List Output		e e	Open Folder in File Explorer		
🗇 Ready	Ln 1 Col 1	Ch 1	Properties	Alt+Enter	🔶 Add to Source Control 🔺

- **12.** Upon successful build modified libAODV.dll file gets automatically updated in the directory containing NetSim binaries.
- **13.** Then Secure_AODV_Workspace comes with a sample configuration that is already saved. To open this example, go to Open Simulation and click on the Secure_AODV_Example that is present under the list of experiments as shown below:

NetSim Home					-		×
NetSim Standard Network Simulation/Emulation Platfor Version 11.1.11 (32 Bit)	orm				www	NETCOS.CC	om
	Current workspace: Secure_AODV_Wo	rkspace		C Experiment name			
New Simulation Ctrl+N	Experiment name	Date modified	Network type				
Open Simulation Ctrl+O	Secure_AODV_Example	27-03-2019	MANET	View Results	Export	Ŵ	
Examples License Settings							
	Workspace options				Import Exper	iment	
Support	Learn		Documentation	Contact	us		
Answers/FAQ Contact Technical Support Email - support@tetcos.com	Videos Experiments Manual		User Manual Technology Libraries Source Code Help	Email - s Phone -	ales@tetcos.co +91 767 605 43	m 321	

14. Secure AODV logs **Secure_AODV.txt** file in the bin folder present in NetSim's installed directory. This can be explained in next section

USE CASES:

1. Secure AODV implementation

Here users can enable Secure AODV (Open AODV.h file)



A Secure_AODV.c file is added to the AODV project which contains the following important functions:

saodv_encrypt_packet()

This function is used to encrypt the control packet data

saodv_decrypt_packet()

This function is used to decrypt the control packet data

get_rrep_str_data()

This function is used to get the route reply data from AODV_RREP control packet

get_rreq_str_data()

This function is used to get the route request data from AODV_RREQ control packet

get_saodv_ctrl_packet_type()

This function is used to change the control packet type from AODV (AODV_RREQ, AODV_RREP) to SAODV (SAODV_RREQ, SAODV_RREP)

get_saodv_ctrl_packet()

This function is called whenever a new control packet is generated

get_aodv_ctrl_packet()

This function is called while processing the control packets

Comment the line #define MALICIOUS_ENABLE present in AODV.h file. Rebuild the solution and replace the dlls as explained before and run the simulation.

After simulation of the given Configuration file, open packet animation. In the packet users can notice **SAODV_RREQ** and **SAODV_RREP** control packets.



The SAODV codes also logs certain details in SAODVlog.txt. The format of the log file is such that each control packet is logged. The first line represents the packet type and the numbering used in a NetSim internal numbering system where by **30701 is RREQ and 30702 is RREP**. The second line is the message which is encrypted. The third line contains the encrypted message after running the RSA encryption algorithm. The fourth line is after decryption and if everything is OK, the 2nd and 4th lines must match

Packet Type = 30701

Org Data = 1,0,1,11.1.1.6,0,11.1.1.1,1

Encrypted Data = *-Ÿ-*-**;*;*;-Ÿ-**;*;*;*;*

Decrypted Data = 1,0,1,11.1.1.6,0,11.1.1.1,1

.....

2. Malicious node implementation

Here users can enable code to malicious node problem. Enable #define MALICIOUS_ENABLE and comment #define SAODV_ENABLE that are present inside AODV.h file.

Solution Explorer $ wedge q imes heta$	AODV.h + ×		
© © ∰ To - 5 a "	🔄 AODV	- (Global Scope)	•
Search Solution Explorer (Ctr 👂 -	25		
Solution 'NetSim' (I project)	26		
	27	//#define SAODV_ENABLE	
Beferences	28	#define MALICIOUS_ENABLE	
External Dependencies	29		
AODV.c	30		
AODV.h	31	#define AODV_ACTIVE_ROUTE_TIMEOUT	3000* MILLISECOND
++ AODV_CheckRouteFou	32	#define AODV_ALLOWED_HELLO_LOSS	2
++ AODV_RouteError.c	33	#define AODV_BLACKLISI_IIMEOUI	AODV_RREQ_RETRIES * AODV_NET_TR
++ FIFOBuffer.c	34	#define AODV_DELETE_PERIOD	K * max (AODV_ACIIVE_ROUIE_IIME
t+ GeneralPacketProcessir in in in in in in in in in in in in in in in in in in in in in in in in in in in in	35	#define K 5	
++ HelloMessage.c	36	#define AODV_HELLO_INTERVAL	1000* MILLISECOND
P **+ Malicious.c	37	#define AODV_LOCAL_ADD_IIL	
v ** RouteViaint.c	38	#define AODV_MAX_REPAIR_IIL	0.3 * AODV_NET_DIAMETER
b ++ RRFP.c	39	#define AODV_MIN_REPAIR_IIL	//see note below
▶ ++ RREQ.c	40	#define AODV_MT_ROUTE_TIMEOUT	2 * AUDV_ACTIVE_ROUTE_TIMEOUT
▶ ++ RSA.c	41	#define AODV_NET_TRAVERSAL_TIME	
++ Secure_AODV.c	42	#define AODV_NEXT_HOP_WAIT	AODV_NODE_TRAVERSAL_TIME + 10

Malicious node advertises wrong routing information to produce itself as a specific node and receives whole network traffic.

After receiving whole network traffic it can either modify the packet information or drop them to make the network complicated

In packet animation, users can notice that malicious node will take all the packets and drops without forwarding to destination

A file **malicious.c** is added to the AODV project which contains the following functions:

IsMaliciousNode ()

This function is used to identify whether a current device is malicious or not in-order to establish malicious behavior.

fn_NetSim_AODV_MaliciousRouteAddToTable()

This function is used to add a fake route entry into the route table of the malicious device with its next hop as the destination.

fn_NetSim_AODV_MaliciousProcessSourceRouteOption()

This function is used to drop the received packets if the device is malicious, instead of forwarding the packet to the next hop.

You can set any device as a malicious node and you can have more than one malicious node in a scenario. Device id's of malicious nodes can be set using malicious_node [] array present in malicious.c file. Comment the line #define SAODV_ENABLE present in AODV.h file. Rebuild the solution and replace the dlls as explained before and run the simulation. If we run simulation without SAODV, we will get zero throughput because malicious node gets all the packets and drops without forwarding to destination. You can notice this in NetSim packet animation.



3. Both Secure AODV and Malicious node implementation

Enable the below mentioned lines of code present in AODV.h file.

#define SAODV_ENABLE #define MALICIOUS_ENABLE

Rebuild the solution and replace the dlls as explained before and run the simulation. Packets will be transmitted to the destination, since SAODV helps in overcoming the Malicious Node problem. Route reply RREP from malicious node 4 will not be accepted by Node 1. It takes the Route reply from node 2 and forms the route.

The SAODV logs certain details in **Secure_AODV.txt**. The first line represents the packet type 30701 = RREQ. The second line is the message logged by SAODV when malicious node tries to decrypt the message

.....

Packet Type = 30702

Encryption and decryption fails. This could be a malicious node

.....

Packet Type = 30702

Encryption and decryption fails. This could be a malicious node

Code modifications done:

Please note that in this project we have added Secure_AODV.c, RSA.c and Malicious.c files

We have added the following macros in AODV.h file

#define SAODV_ENABLE
#define MALICIOUS_ENABLE

Then we have added the following lines of code in enum_AODV_Ctrl_Packet in AODV.h file

//#ifdef SAODV_ENABLE
 SAODV_RREQ,
 SAODV_RREP,
 SAODV_RERR,
//#endif

Then we have added the following function prototypes in AODV.h file

```
#ifdef SAODV_ENABLE
    void get_saodv_ctrl_packet(NetSim_PACKET* packet);
    void get_aodv_ctrl_packet(NetSim_PACKET* packet);
    void saodv_copy_packet(NetSim_PACKET* dest, NetSim_PACKET*
src);
    void saodv_free_packet(NetSim_PACKET* packet);
    void remove_from_mapper(void* ptr,bool isfree);
#endif
```

bool IsMaliciousNode(NETSIM_ID devId);

We have added the following function prototypes in AODV.c file

```
bool IsMaliciousNode(NETSIM_ID devId);
int fn_NetSim_AODV_MaliciousRouteAddToTable(NetSim_EVENTDETAILS*);
int
fn_NetSim_AODV_MaliciousProcessSourceRouteOption(NetSim_EVENTDETAILS
*);
```

Then we have added the following lines of code in NETWORK_IN event in fn_NetSim_AODV_Run() function present in AODV.c file

```
#ifdef SAODV_ENABLE
    switch(pstruEventDetails->pPacket->nControlDataType)
    {
        case SAODV_RREQ:
        case SAODV_RREP:
        case SAODV_RERR:
            get_aodv_ctrl_packet(pstruEventDetails->pPacket);
            break;
        }
        if(pstruEventDetails->pPacket == NULL)
        {
            return -1; //Decryption fail.
        }
    }
}
```

#endif

We have added the following lines of code in AODVctrlPacket_RREQ and default cases in NETWORK_IN event to check the current node is malicious or not

if(IsMaliciousNode(pstruEventDetails->nDeviceId))

fn_NetSim_AODV_MaliciousProcessSourceRouteOption(pstruEventDet
ails);

Then we have added the following code in fn_NetSim_AODV_CopyPacket () function present in AODV.c file

```
#ifdef SAODV_ENABLE
    switch(srcPacket->nControlDataType)
    {
        case SAODV_RERR:
        case SAODV_RREQ:
        case SAODV_RREP:
            saodv_copy_packet(destPacket,srcPacket);
            return 0;
            break;
```

```
default:
#endif
return fn_NetSim_AODV_CopyPacket_F(destPacket,srcPacket);
#ifdef SAODV_ENABLE
break;
}
#endif
```

Then we have added the following code in int fn_NetSim_AODV_FreePacket () present in the AODV.c file

```
#ifdef SAODV ENABLE
      switch(packet->nControlDataType)
      {
      case SAODV_RERR:
      case SAODV_RREQ:
      case SAODV RREP:
             saodv_free_packet(packet);
             return 0;
             break;
      default:
             remove_from_mapper(packet->pstruNetworkData-
>Packet_RoutingProtocol, true);
             return 0;
             break;
       }
#endif
```

Then we have added the following function calls in fn_NetSim_AODV_GenerateRREQ (), fn_NetSim_AODV_RetryRREQ () and fn_NetSim_AODV_ForwardRREQ () functions present in RREQ.c file

Then we have added the following function calls in fn_NetSim_AODV_GenerateRREP(), fn_NetSim_AODV_ForwardRREP () and fn_NetSim_AODV_GenerateRREPByIntermediate () functions present in RREP.c file

```
#ifdef SAODV_ENABLE
    get_saodv_ctrl_packet(packet);
#endif
```