AODVDC: An Improved Protocol Prevents Whirlwind Attacks in Mobile Ad hoc Network

Luong Thai Ngoc[1,2], Vo Thanh Tu[1]

(Corresponding author: Luong Thai Ngoc)

Faculty of Information Technology, Hue University of Sciences, Hue University; Viet Nam[1]
77 Nguyen Hue street, Hue city, Vietnam
Faculty of Mathematics and Informatics Teacher Education, Dong Thap University; Viet Nam[2]
783 Pham Huu Lau street, Ward 6, Cao Lanh city, Dong Thap, Viet Nam
(Email: ltngoc@dthu.edu.vn, vttu@hueuni.edu.vn)

(Received Oct. 28, 2017; revised and accepted Apr. 13, 2018)

Abstract

Ad hoc On-demand Distance Vector routing protocol is one of the most popular reactive protocol used for Mobile Ad hoc Network, is target of many denial-of-Service attack types. Whirlwind attacks uses a malicious node to make one routing-loop on the discovered route. All data packets are dropped due to they over time-life. This article proposes a mechanisms to manage and provide digital certificates (DC) for Mobile Ad hoc Network (MANET) without public key infrastructure. A digital certificates authentication mechanism secure that only "friendly" nodes to collaborate in the route discovery process, goal is to prevent malicious nodes that joined the discovered route, such as Whirlwind. A new routing protocol named AODVDC by integrating our solutions into AODV routing protocol. Using NS2, we evaluate the security performance using scenario where there are nodes move ramdomly and Whirlwind attacks, compared with related protocols. The simulation results showed that our approach has better performance in terms of packet delivery ratio, routing load and route discovery delay compared to related works under attack scenario.

Keywords: AODV; AODVDC; MANET; Network Security; Whirlwind Attacks

1 Introduction

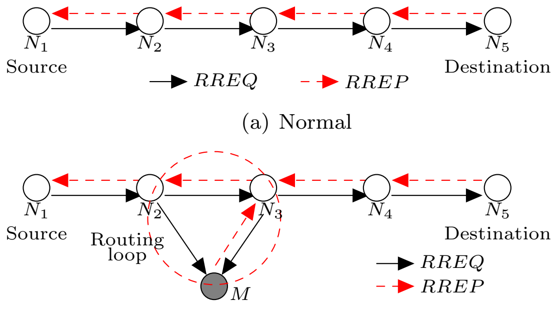
Mobile Ad hoc Network (MANET [8]) is a special wireless, the advantages such as flexibility, mobility, every mobile node acts both as a host and as a router. Routing is the main service provided in network layer, the source node using the route to the destination is discovered and maintained. There are many routing protocols are recommended to MANET, they are classified into proactive, reactive, and hybrid routing [2]. Ad hoc On-demand Distance Vector (AODV [16]) routing protocol is one of the most popular reactive protocol used for Mobile Ad hoc Network, is target of many denial-of-Service attack types [17], such as Blackhole [4, 9], Sinkhole [5], Grayhole [7], Flooding [20], Wormhole [3] and Whirlwind [15].

We focus on Whirlwind attacks type and prevention solution, this attack type target is to make routing-loop which is done with two phases:

Phase 1: Malicious nodes try to set up a routing-loop in the discovered route from source to destination node when receiving route request packet (RREQ) from any source node N_S by using the fake route reply packet (FRREP).

Phase 2: If attacking is successful, all data packets from source to destination node are taken into data whirlwind and automatically dropped due to over time-life.

In Figure 1(a), source node N_1 discovers a new route to destination node N_5 by broadcasting of RREQ to its neighbor nodes named N_2. Intermediate node N_2 is not destination node, it therefore continue broadcasts RREQ packet to its neighbors named N_3 and save reserve route to source N_1, this process repeats at N_3 and N_4 until node N_5 receives the route request packet. When receiving RREQ packet from node N_4, destination node N_5 sends unicast of RREP packet to source on route $\{N_5 \rightarrow N_4 \rightarrow N_3 \rightarrow N_2 \rightarrow N_1\}$. As a result, source node N_1 discovers route to destination in following direction $\{N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_4 \rightarrow N_5\}$. Figure 1(b) shows that malicious node M appears in network topology for Whirlwind attack behavior, it is neighbor of both N_2 and N_3 nodes. When receiving the first RREQ packet from node N_2, M adds a entry to destination into its routing table (RT) with minimum cost and next hop (NH) is N_2. When receiving the second RREQ packet from N_3, M adds a entry to source N_1 into its RT with lowest cost and NH is N_3, concurrently sends unicast of FRREP to source N_1 in direction $\{M \rightarrow N_3 \rightarrow N_2 \rightarrow N_1\}$. As a result, N_3's RT has route information to destination via *NH* is M with the lowest cost. The destination node N_5 also sends a RREP packet to source node on direction $\{N_5 \rightarrow N_4 \rightarrow N_3 \rightarrow N_2 \rightarrow N_1\}$. When receiving the RREP packet from node N_4, node N_3 see that the cost to destination is not cheaper than the existing route, the RREP packet is therefore dropped. The results is exist routing-loop on discovered route from N_1 to N_5 including nodes named N_2, N_3, and M. All data packets from N_1 to N_5 node are taken into data whirlwind and automatically dropped due to over time-life.



(a) Normal

(b) Attacks

Figure 1: Description of whirlwind attacks [15]

This article proposes a mechanisms to manage and provide digital certificates for MANET without public key infrastructure (PKI) because MANET is no infrastructure. In addtion, digital certificates authentications mechanism allows only "friendly" node to collaborate in the route discovery process. The remainder of this article is structured as following: In the next Section, we review some related works for security base on digital signature. Section 3 mechanism to manage and provide the digital certificate. Section 4 shows how to authenticate preceding node's DC when an node receiving the control route packets. Section 5 shows the evaluation results by simulation; Finally, conclusions and future works.

2 Related Works

There are some related works to increase security level for AODV routing protocol based on digital signature or one-way hash [12]. Zhou [24] described a solution to distribute the CA role among n nodes of the network using *(n, k + 1)* threshold cryptography scheme. In this scheme the private key is divided into n partial shares $(S_1, S_2, \cdots, S_n)$ where at least*k+1* of n are partial shares which are needed to generate a secret S. The advantage is its increased availability, since any *k+1* among n nodes in the local neighborhood of the requesting node can issue or renew a certificate. And any node, which does not have a private share yet, can obtain a share from any group of at least *k + 1* nodes which has already a share [1].

Zhang described a solution named IKM (id-based key management) as a novel combination of ID-based and threshold cryptography. IKM is a certificateless solution in that public keys of mobile nodes are directly derivable from their known IDs plus some common information. It thus eliminates the need for certificate-based authenticated public-key distribution indispensable in conventional public-key management schemes. IKM features a novel construction method of ID-based public/private keys, which not only ensures high-level tolerance to node compromise, but also enables efficient network-wide key update via a single broadcast message [23].

Zapata in [22] recommended SAODV is improved from AODV to prevent impersonation attacks by changing hop-count (HC) and sequence number (SN) values of route discovery packet. However, SAODV only supports authentication from end-to-end without authenticating hop-by-hop, hence, intermediate node can't certify packet from the preceding node. Addition, because SAODV does not have a mechaism for authentication intermediate node and public key management, malicious nodes can easily join a route by using fake keys.

Sanzgiri [18] recommended ARAN protocol, different from SAODV, route discovery packet (RDP) in ARAN is signed and certified at all nodes. ARAN supplemented the testing member node mechanism, thus, malicious can not pass over security by using fake keys. Structure of RDP and reply route (REP) packets of ARAN is not available with HC to identify routing cost; this means ARAN is unable to recognize transmission expenses to the destination, ARAN argued that the first REP received is the route packet with the best expenses.

Li [10] recommended SEAODV using certification scheme HEAP with symmetric key and one-way hash function to protect route discovery packet. By simulation, the authors has shown that SEAODV is more security with lower communication overhead.

3 Digital Certificates Management and Providing Model

This section describes the digital certificates structure based X.509 and mechanism to manage and provide the Digital Certificate for MANET without PKI. For this approach, we assumptions that each node has a unique identifier and a pair of keys: a private key and a public key. Set of symbols in Table 1 are applied for the presentation.

Table 1: Description of symbols

Variable	**Descriptions**
N_δ	Node labeled δ
$k_{N_\delta}+$, $k_{N_\delta}-$	Public and private keys of node N_δ
$En(v,k)$	Encrypting v using key k
$De(v,k)$	Decrypting v using key k
$H(v)$	v is hashed by SHA_1 [14] function
IP_{N_δ}	Address of node N_δ
DC_{N_δ}	Digital Certificate of node N_δ

3.1 Digital Certificates

Digital certificates are used to certify the identities of nodes in MANET, it is provided for node automatically from certificate authorities (CA) before nodes collaborate to the discovery route process. We uses a X.509 certificate template, has the structure as Figure 2. Where,

1. Version
2. Serial Number
3. Signature Algorithm
4. Issuer Name
5. Validity Period
6. Subject Name
7. Public Key (PK)
8. Certificate Signature (CS)

Figure 2: DC structure based on X.509 [13]

1) Certificate version;

2) The unique serial number that is assigned by the CA;

3) The public key cryptography and message digest algorithms that are used by CA;

4) The name of the issuing CA;

5) The certificate's start and expiration dates. These define the interval during which the certificate is valid, although the certificates can be revoked before the designated expiration date;

6) The name of the subject of the certificate;

7) The public key and a list of the public key cryptography algorithms;

8) The CA's digital signature, which is created as the last step in generating the certificate by encrypting the hash value of all X.509 certificates attributes with of CA private keys as Formula 1.

$$CS \leftarrow En(H(DC.AllFields \backslash \{CS\}), k_{N_{CA}}-). \qquad (1)$$

Algorithm 1 shows steps to authenticate DC of the packet RREQ (or RREP) if N_i node receiving the packet from preceding node N_j. Node N_i uses the public key ($k_{N_{CA}}+$) of certificate authorities to decrypt the CS field value of packet RREQ (or RREP). If the value after decryption is coincident with the hash value of all fields (excepted CS) for DC then DC is valid, on the contrary then DC is invalid.

3.2 Digital Certificate Management

We setup a reliable node named N_{CA} acts as certificate authorities to provide DC for all member nodes. In N_{CA} exists a Digital Certificate Database (DCDB) of all nodes as Table 2. Each record in DCDB consists of: Nodes address, OK field controlling the node certificated with DC and its Digital Certificates. Where, all attributes (except OK field) are updated by administrators to ensure that only "friendly" nodes are provided with DC.

Algorithm 1 Checking Digital Certificate

Input: RREQ or RREP packet; **Output:** True if DC is valid; Else return False

```
Boolean IsValidDC(Packet P)
Begin
    $val_1 \leftarrow De(P.DC.CS, k_{N_{CA}}+)$;
    $val_2 \leftarrow H(P.DC.AllFields \backslash \{CS\})$;
    Return ($val_1 == val_2$);
End
```

Table 2: Digital certificate database

Nodes	OK	Digital Certificate
IP_{N_1}	yes	DC_{N_1}
IP_{N_2}	yes	DC_{N_2}
IP_{N_3}	no	DC_{N_3}
...	...	...
IP_{N_n}	yes	DC_{N_n}

3.3 Digital Certificate Providing

We propose a digital certificate providing model which secure that

1) Malicious node can not action as CA node to provide DC to member node;

2) Only the valid member node receives the DC from CA node.

There are two DCP and DC_{ACK} packets are used to provide the Digital Certificates for all nodes. They have the structures similar as RREQ and RREP packets, DCP packet has a new field named DC to store the digital certificate, DC_{ACK} has two new fields named ACK and KEY, they save acknowledge information and public key from member node. The steps to provide the DC for all nodes following:

- *The first*, administrators update DC of "friendly" nodes to DCDB. Member nodes can not to collaborate in the route discovery process until they have received DC from N_{CA}.

- *The second*, periodically after T_{DC} time interval, node N_{CA} checks all nodes are provided with DC by using the DCDB information. If exist node N_δ that it is not provided with DC (OK = False), N_{CA} broadcasts the DCP packet to provide the DC for N_δ.

- *Continuous*, when receiving DC, node N_δ sends DC_{ACK} packet back to N_{CA} to confirm that member node already receives DC if DCP packet is sent by N_{CA} and sent for it.
- *Finally*, when receiving packet DC_{ACK}, N_{CA} checks: If the packet is sent by N_δ, N_{CA} updates OK value is true to DCDB, else this process is fail.

3.3.1 Broadcasting DCP Packet and Saving DC

Node N_{CA} provides a DC for node N_δ by broadcasting DCP packet, is improved from algorithm broadcasting RREQ packet of AODV following:

1) *Generating DCP packet:* Node N_{CA} creates DCP with DC_{N_δ} and broadcasts it to all its neighbors as Formula 2.

$$N_{CA}broadcasts : DCP \leftarrow \{RREQ^* + DC_{N_\delta}\}. \quad (2)$$

Where $RREQ^*$ is the original RREQ packet of AODV protocol and DC is N_δ 's Digital Certificate. CS field value in DC that it is calculated as Formula 3.

$$DC.CS \leftarrow En(DC.CS, k_{N_\delta}+). \quad (3)$$

2) *Checking DCP and saving DC:* When node N_δ receives the DCP packet, it tests that DCP is sent by N_{CA} and provided DC for N_δ. If all the conditions are satisfied, N_δ saves DC into its cache and unicasting the DC_{ACK} packet to confirm for N_{CA}. On contrary, the packet is dropped, see in Algorithm 2.

Algorithm 2 Testing and Saving Digital Certificate;

Input: DCP packet; **Output:** True if DC is saved successful; Else return False;

```
1:  Boolean TestAndSaveDC(DCP P)
2:  Begin
3:      $val_1 \leftarrow De(P.DC.CS, k_{N_\delta}-)$;
4:      $val_2 \leftarrow De(val1, k_{N_{CA}}+)$;
5:      If $val_2 != H(P.DC.AllFields \backslash \{CS\})$ Then
6:          Dispose(P) and Return False;
7:      Else
8:          $P.DC.CS \leftarrow val1$;
9:          SaveToCache(P.DC);
10:         Sends $DC_{ACK}$ packet back to $N_{CA}$;
11:         Return True;
12: End
```

We clearly see that malicious nodes can easily receive DCP packet come from the N_{CA} node because they are sent in the form of a broadcast. However, the malicious node can not decrypt the contents of the certification in DC of DCP packet because it does not know the secret key of N_δ node. If exists any change in the DC packet resulting in command 5 in Algorithm 2 is true, the DCP packet is canceled, the DC proveding process is fail.

3.3.2 Replying the DC_{ACK} Packet

Member node N_δ sends a DC_{ACK} packet back to confirm for N_{CA}, this algorithm is improved from unicasting RREP packet algorithm of AODV following:

1) *Generating DC_{ACK} packet:* After saving DC successfully, node N_δ unicasts confirmation packet DC_{ACK} to back N_{CA} as Formula 4.

$$N_\delta unicasts : DC_{ACK} \leftarrow \{RREP^* + ACK + KEY\} \quad (4)$$

Where $RREP^*$ is the original RREP packet of AODV routing protocol and ACK field is calculated by Formula 5, KEY field value is its public key.

$$DC_{ACK}.ACK \leftarrow En(En(H(IP_{Nca}), k_{N_\delta}-), k_{Nca}+)) \quad (5)$$

2) *Checking DC_{ACK} and updating DCDB:* When node N_{CA} receives the DC_{ACK} packet, it tests DC_{ACK} packet is sent by N_δ and it is target node. If all the conditions are satisfied, N_{CA} updates successfully provided DC to DCDB, on contrary, the packet is dropped, see in Algorithm 3.

Algorithm 3 Testing DC_{ACK} and Updating DCDB

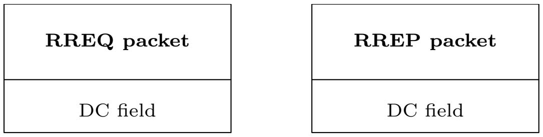
Input: DC_{ACK} packet; **Ouput:** True if DC is provided successful; Else return False

```
Boolean TestDC$_{ACK}$($DC_{ACK}$ P)
Begin
    $val_1 \leftarrow De(P.ACK, k_{N_{CA}}-)$;
    $val_2 \leftarrow De(val1, P.KEY)$;
    If $val_2 != H(IP_{Nca})$ Then
        Dispose(P) and Return False;
    If ($IP_{N_\delta}$ exists in DCDB) Then
        DCDBRow row $\leftarrow$ DCDB.Rows[$IP_{N_\delta}$];
        row.OK $\leftarrow$ True;
        Return True;
    Else
        Dispose(P) and Return False;
End
```

We clearly see that a malicious node can hardly receive DC_{ACK} packet because this packet is sent in unincast form. Moreover, malicious nodes can not be act as N_δ to send DC_{ACK} packet to N_{CA}. The reason is because it does not have the secret key of N_δ, and the public key of N_δ was administered by N_{CA}.

4 AODVDC: Improved Protocol Using Digital Certificates

An algorithm has been designed based on reactive routing protocols accepted as standards for routing in MANETs such as AODV. However, the AODV protocol have not any security mechanism for discovery route processing. This is the hole which can be easily exploited by hackers to attack the network by modifying the control packets with fake information. Improved protocol named AODVDC, is proposed by integration of DC authentication algorithm into AODV protocol includes the two phases: Broadcasting route request packet and unicasting route reply packet. The control route packet structures of new protocol is improved from control route packets in AODV, they are supplemented a new field named DC as Figure 3.



(a) New RREQ packet (b) New RREP packet

Figure 3: Control packet structures of AODVDC

4.1 Broadcasting Route Request Packet

Figure 4 describes route request algorithm using DCa method, it is improved from AODV route discovery algorithm as following:

1) *Generating RREQ packet:* If source node (N_S) has not a route to destination node, it starts a new route discovery process by broadcasting the RREQ packet to its all neighbors described as Formula 6.

$$N_S broadcasts : RREQ^* + DC_{N_S} \tag{6}$$

Where $RREQ^*$ is the original RREQ packet of AODV routing protocol and DC is its Digital Certificate.

2) *Processing and forwarding RREQ packet:* When a node receiving a RREQ packet, intermediate or destination node (N_i) processes the packet following:

   - If it has not the DC Then N_i drops RREQ packet and The end;
   - Else, N_i tests the preceding node 's DC in RREQ packet using *IsValidDC()* function. If DC is invalid Then N_i drops the RREQ packet due to discovered route has malicious node and The end;
   - Else, if current node is the destination, it just simply generates and sends back the RREP packet and The end;
   - Else, it updates a reverse route toward the source node and updates the RREQ packet using its information and DC before continuous broadcasting the RREQ packet to its all neighbors.

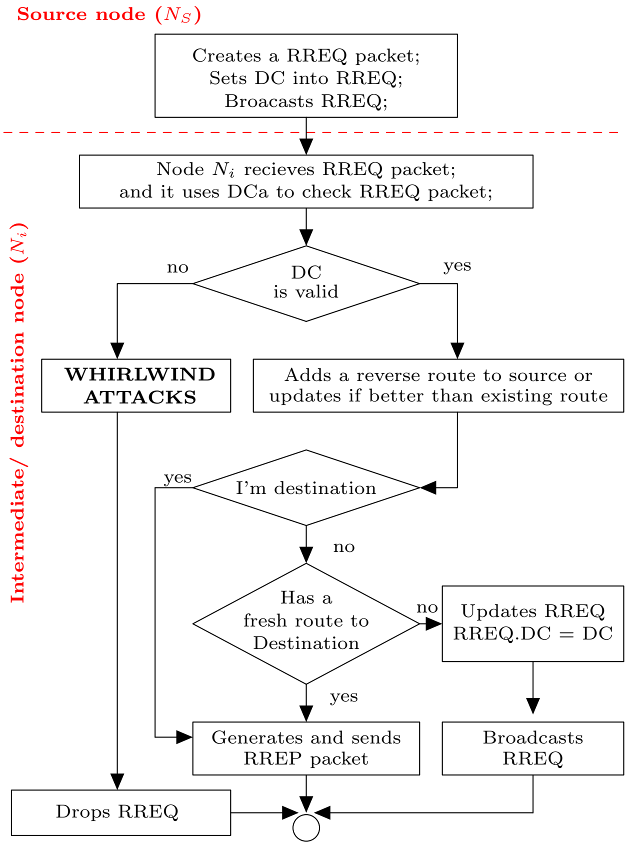


Figure 4: Improved request route algorithm

4.2 Unicasting Route Request Packet

Figure 5 describes route reply algorithm using DCa method, it is improved from AODV route reply algorithm as following:

1) *Generating RREP packet:* A node generates the RREP packet if it is either the destination (N_D) or an intermediate (N_i) which has a "fresh" route to the destination described as Formula 7.

$$N_D unicasts : RREP^* + DC_{N_D} \tag{7}$$

Where $RREP^*$ is the original RREP packet of AODV routing protocol and DC is its Digital Certificate.

2) *Processing and forwarding RREP packet:* When a node receiving a RREP packet, intermediate or destination node (N_i) processes the packet following:

- If it has not the DC Then N_i drops this packet and The end;
- Else, N_i tests the preceding node 's DC in RREP packet using *IsValidDC()* function. If DC is invalid Then N_i drops the RREP packet due to discovered route has malicious node and The end;
- Else, if current node is the source node, it just simply saves a new route or updates if better than existing

route and send data packets from queue to the destination node through discovered route;

- Else, it saves a route to the destination node and updates the RREP packet using its information and DC before continuous unicasting the RREP to back source node.

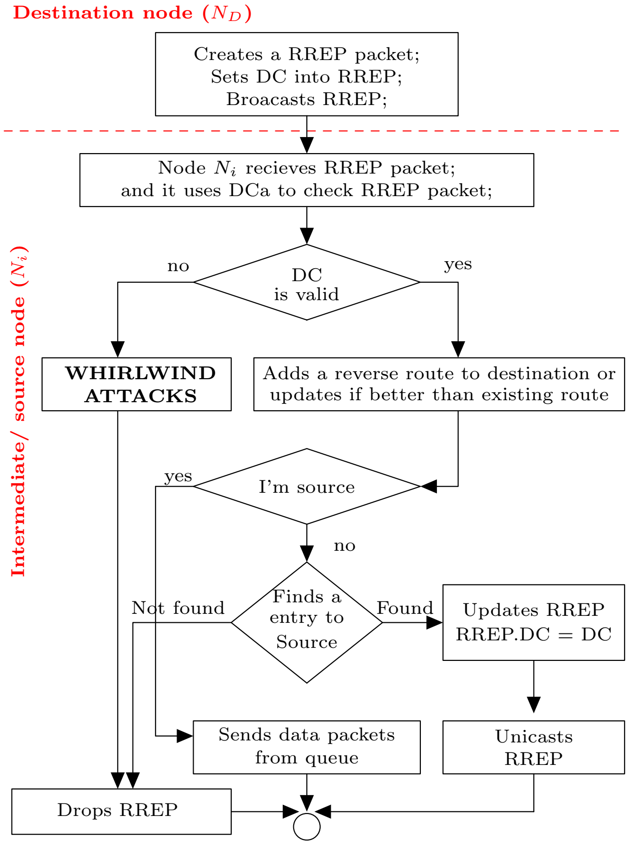

Destination node (N_D)
Creates a RREP packet; Sets DC into RREP; Broacasts RREP;
Intermediate/ source node (N_i)
Node N_i recieves RREP packet; and it uses DCa to check RREP packet;
DC is valid
no
yes
WHIRLWIND ATTACKS
Adds a reverse route to destination or updates if better than existing route
I'm source
yes
no
Finds a entry to Source
Not found
Found
Updates RREP RREP.DC = DC
Sends data packets from queue
Unicasts RREP
Drops RREP

Figure 5: Improved reply route algorithm

5 Simulation Results

We evaluate the Whirlwind attacks prevention performance of AODVDC on simulation system is NS2 - version 2.35 [11]. The simulation area was a rectangular region with a size of 2000 x 2000 m^2, which was chosen to ensure that there existed multiple hops within the network. We use 802.11 MAC layer, 100 normal nodes move with 30m/s maximum speeds under Random Waypoint [21] model, 1 malicious node stays at the center position (red rectangle in Figure 6) and starts to attack at 500s.

Each scenario has 10 pairs of communicating nodes, source sending out constant bit rate (CBR) traffic with packet sizes of 512bytes, rate of 2 packet per second. The first data source is started at second of 0, the following data source is 5 seconds apart from each node. Time 1000 seconds for simulation, FIFO queue type, the detail of simulation parameters are listed in the Table 3.

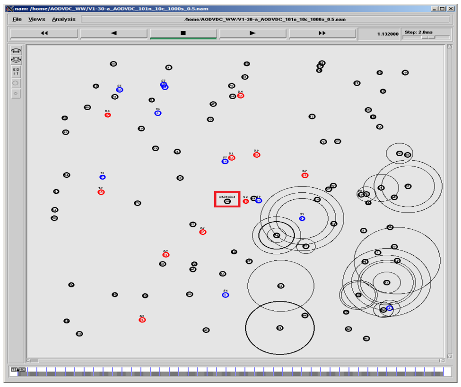


Figure 6: NS2 simulation screen

Table 3: Simulation parameters

Parameters	Setting
Simulation area (m)	2000 x 2000
Simulation time (s)	1000
Number of nodes	101 (1 malicious nodes)
Attack point-time (s)	500^{th}
Wireless standard	IEEE 802.11
Ratio range (m)	250
Mobility model	Random Waypoint
Mobility speed (m/s)	1..30
Number of connection	10 UDPs
Traffic type	CBR
Data rate	2 pkt/s (512 bytes/pkt)
Queue type	FIFO (DropTail)
Routing protocols	AODV, ARAN and AODVDC
Nca	N_{50}

Some used metrics for evaluation following: Packet overhead for providing DC, packet delivery ratio (PDR), routing load (RL) and end-to-end delay (EtE).

5.1 Packet Overhead for Providing DC

We analyse the packet overhead (DCP and DC_{ACK}) for providing the DC in normal network topology. The first scenario simulates for 100 nodes used AODVDC protocol, all DC of normal nodes are setup in DCDB; The second scenario simulates for 100 nodes used AODVDC protocol with 80 normal nodes from 0 to 79 identify in DCDB; The final scenario, we use a scenario similar to the second scenario, and 20 new nodes are installed into DCDB at 300th seconds. The simulation results in Figure 7 shows that AODVDC needs total 57,908 packets DCP and DCACK overhead and 560s to provide DC for all 100 nodes. For the second scenario, there are total 30,096 packets overhead and providing DC has finished during 200s. In the final scenario, there are 36,253 packets overhead and 420s for finished providing DC.

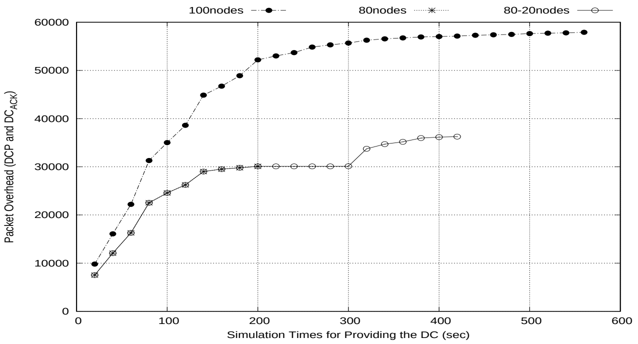


Figure 7: Packet overhead for providing DC

5.2 Whirlwind Attack Prevention Performance

The main purpose for whirlwind attack is to destroy data packets, reduced packet delivery ratio. Figure 8 shows that packet delivery ratio of AODV go down significantly under whirlwind attacks, reduced during simulation times from seconds 500^{th}. The packet delivery ratio of AODVDC increasing from seconds 600^th because it uses first 560 seconds for providing DC for member nodes. After 1000s for simulation with 10UDP connections, the packet delivery ratio of AODV is 71.04% for normal network topology down to 58.02% under whirlwind attacks, reduced 13.02%. The ARAN packet delivery ratio is 59.51% and AODVDC is 65.49%. It is then clear that the AODVDC packet delivery ratio is improved significantly and has better packet delivery ratio compared to ARAN.

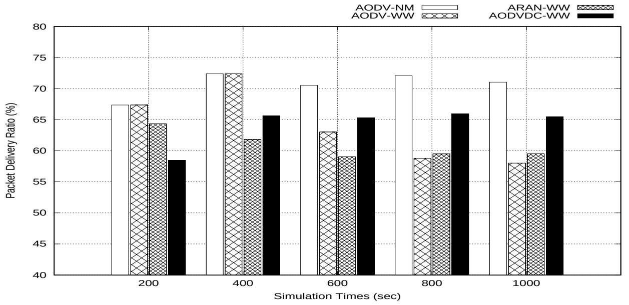


Figure 8: Packet delivery ratio; WW: Whirlwind, NM: Normal

5.3 Routing Load

The AODVDC routing load is larger than original one due to proposed approach used overhead packets DCP and DC_{ACK} for providing the DC for all member nodes. Figure 9 shows that the routing load of AODVDC is larger than AODV, reduced during simulation times due to finished providing DC for member nodes. After 1000s for simulation with 10UDP connections, the routing load of AODVDC is 22.37pkt, AODV is 17.74pkt and ARAN is 20.05pkt.

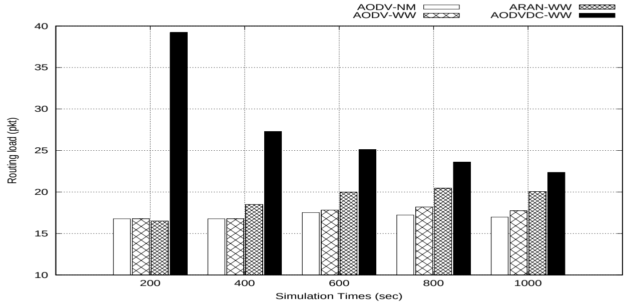


Figure 9: Routing load; WW: Whirlwind, NM: Normal

5.4 End-to-End Delay

Figure 10 shows that all security protocols have end-to-end delay is higher than AODV because of they used RSA public key encryption and hash function SHA_1 for security goal. After 1000s for simulation, end-to-end delay of AODV is 0.867s, ARAN is 1.214s and AODVDC is 1.279s.

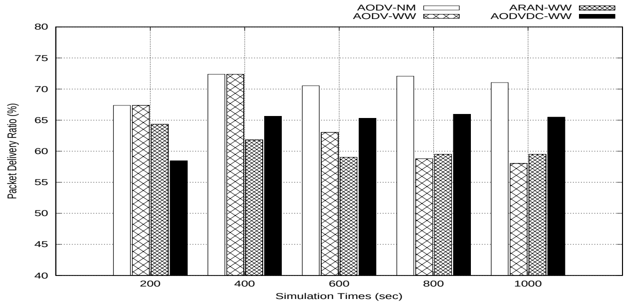


Figure 10: End-to-End delay; WW: Whirlwind, NM: Normal

6 Conclusion

We proposed a mechanisms to manage and provide digital certificates (DC) for Mobile Ad hoc Network (MANET) without public key infrastructure. A new routing protocol named AODVDC by integrating our solutions into the discover route process from AODV protocol. The simulation results showed that our approach has better performance in terms of packet delivery ratio, routing load and route discovery delay compared to related works under attack scenario. However, AODVDC has routing load and end-to-end delay are larger than AODV because it uses new control packets for providing DC for member nodes and uses RSA [6] public key encryption, SHA_1 [14] hashing function.

In the future, we will setup AODVDC with large key to improve the security performance using TLS library [19] and comparison with related works.

References

[1] R. Abderrezak, B. Abderrahim, "A secure architecture for mobile Ad hoc networks," *Mobile Ad-hoc and Sensor Networks*, pp. 424–435, 2006. (https://link.springer.com/chapter/10.1007\%2F11943952_36)

[2] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-hoc and mesh networks," *Computer Networks*, vol. 56, no. 2, pp. 940–965, 2012. (http://www.sciencedirect.com/science/article/pii/S138912861100377X)

[3] A. P. Asad and C. McDonald, "Detecting and evading wormholes in mobile Ad-hoc wireless networks," *International Journal of Network Security*, vol. 3, no. 2, pp. 191–202, 2006. (http://ijns.jalaxy.com.tw/contents/ijns-v3-n2/ijns-2006-v3-n2-p191-202.pdf)

[4] C. W. Badenhop, B. W. Ramsey and B. E. Mullins, "An analytical black hole attack model using a stochastic topology approximation technique for reactive Ad-hoc routing protocols," *International Journal Network Security*, vol. 18, no. 4, pp. 667–677, 2016. (http://ijns.jalaxy.com.tw/download_paper.jsp?PaperID=IJNS-2015-06-11-1&PaperName=ijns-v18-n4/ijns-2016-v18-n4-p667-677.pdf)

[5] L. S. Casado, G. M. Fernández, P. G. Teodoro and N. Aschenbruck, "Identification of contamination zones for Sinkhole detection in MANETs," *Journal of Network and Computer Applications*, vol. 54, pp. 62–77, 2015. (http://www.sciencedirect.com/science/article/pii/S1084804515000818)

[6] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976. (http://dl.acm.org/citation.cfm?id=2269104)

[7] X. Gao and W. Chen, "A novel gray hole attack detection scheme for mobile Ad-hoc networks," in *IFIP International Conference on Network and Parallel Computing Workshops*, pp. 209–214, 2007. (http://ieeexplore.ieee.org/document/4351486/)

[8] H. Jeroen, M. Ingrid, D. Bart and D. Piet, "An overview of mobile Ad hoc networks: Applications and challenges," *Journal of the Communications Network*, vol. 3, no. 3, pp. 60–66, 2004. (https://biblio.ugent.be/record/317876)

[9] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour and Y. Nemoto, "Detecting blackhole attack on AODV-based mobile Ad Hoc networks by dynamic learning method," *International Journal of Network Security*, vol. 5, no. 3, pp. 338–346, 2007. (http://ijns.jalaxy.com.tw/contents/ijns-v5-n3/ijns-2007-v5-n3-p338-346.pdf)

[10] C. Li, Z. Wang and C. Yang, "SEAODV: A security enhanced AODV routing protocol for wireless mesh networks," *Transactions on Computational Science XI*, vol. 6480, pp. 1–16, 2010. (http://link.springer.com/chapter/10.1007\%2F978-3-642-17697-5_1)

[11] S. McCanne and S. Floyd, *The Network Simulator NS2*. (http://www.isi.edu/nsnam/ns/)

[12] J. V. Mulert, I. Welch, and W. K. G. Seah, "Security threats and solutions in MANETs: A case study using AODV and SAODV," *Journal of Network and Computer Applications*, vol. 35, no. 4, pp. 1249–1259, 2012. (http://www.sciencedirect.com/science/article/pii/S1084804512000331)

[13] M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, RFC 2560, June 1999. (https://tools.ietf.org/html/rfc6960https://tools.ietf.org/html/rfc6960)

[14] National Institute of Standards and Technology, "Secure hash standard," FIPS PUB 180-1, 1995. (https://tools.ietf.org/html/rfc3174)

[15] L. T. Ngoc and V. T. Tu, "Whirlwind: A new method to attack routing protocol in mobile Ad hoc network," vol. 19, no. 5, *International Journal of Network Security*, pp. 832–838, 2017. (http://ijns.jalaxy.com.tw/download_paper.jsp?PaperID=IJNS-2016-06-28-1&PaperName=ijns-v19-n5/ijns-2017-v19-n5-p832-838.pdf)

[16] C. E. Perkins, M. Park and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, pp. 90–100, 1999. (https://tools.ietf.org/html/rfc3561)

[17] R. D. Pietro, S. Guarino, N. V. Verde and J. Domingo-Ferrer, "Security in wireless Ad-hoc networks - A survey," *Computer Communications*, vol. 51, pp. 1-20, 2014. (http://www.sciencedirect.com/science/article/pii/S0140366414002242)

[18] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields and E. M. B. Royer, "A secure routing protocol for Ad hoc networks," in *10th IEEE International Conference on Network Protocols (ICNP'02)*, pp. 78–87, 2002. (http://ieeexplore.ieee.org/document/1181388/)

[19] TLS library, RSA source code. (https://tls.mbed.org/rsa-source-code)

[20] V. T. Tu, L. T. Ngoc, "SMA$_2$AODV: Routing protocol reduces the harm of flooding attacks in mobile Ad hoc network," *Journal of Communications*, vol. 12, no. 7, pp. 371–378, 2017. (http://www.jocm.us/index.php?m=content&c=index&a=show&catid=179&id=1122)

[21] J. Yoon, M. Liu and B. Noble, "Random waypoint considered harmful," *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, vol. 2, pp. 1312-1321, 2003. (http://ieeexplore.ieee.org/abstract/document/1208967/)

[22] M. G. Zapata, "Secure Ad hoc on-demand distance vector routing," *Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 106–107, 2002.

[23] Y. Zhang, W. Liu, W. Lou and Y. Fang, "Securing mobile Ad hoc networks with certificateless public keys," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 386–399, 2016. (`http://ieeexplore.ieee.org/abstract/document/4012650/`)

[24] L. Zhou and Z. J. Haas, "Securing Ad hoc networks," *IEEE Network, Special Issue on Network Security*, vol. 13, no. 6, pp. 24-30, 1999.

Luong Thai Ngoc is with the Faculty of Mathematics and Informatics Teacher Education, Dong Thap University. He received B.E. degree in Computer Science from Dong Thap University in 2007 and MSc degree in Computer Science from Hue University of Sciences in 2014. He is a PhD student in Hue University of Sciences now. His fields of interesting are network routing, analysis and evaluation of network performance, security wireless ad hoc network.

Vo Thanh Tu is an associate professor in the Faculty of Information Technology, Hue University of Sciences, Hue University. He received B.E. degree in Physics from Hue University in 1987 and PhD degree in computer science from Institute of Information Technology, Vietnam Academy of Science and Technology in 2005. His fields of interesting are network routing, analysis and evaluation of network performance, security wireless ad hoc network.