RAD: Reinforcement Authentication DYMO Protocol for MANET

Conference Paper · December 2019

DOI: 10.1109/ICPET.2019.00032

Mohammad R. Ayyad
Al-Quds University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Rushdi A Hamamreh
Al-Quds University
36 PUBLICATIONS **68** CITATIONS
SEE PROFILE

Mohammed Jamoos
Al-Quds University
5 PUBLICATIONS **7** CITATIONS
SEE PROFILE

RAD: Reinforcement Authentication DYMO Protocol for MANET

Rushdi A. Hamamreh
Department of computer engineering
Al-Quds University
Jerusalem, Palestine
rushdi@staff.alquds.edu

Mohammad Ayyad
Department of computer engineering
Al-Quds University
Jerusalem, Palestine
rushdi@staff.alquds.edu

Mohammad Jamoos
Department of computer science
Al-Quds University
Jerusalem, Palestine
jamoos@staff.alquds.edu

***Abstract*: Mobile ad hoc network (MANET) does not have fixed infrastructure or centralized server which manages the connections between the nodes. Rather, the nodes in MANET move randomly. Thus, it is risky to exchange data between nodes because there is a high possibility of having malicious node in the path. In this paper, we will describe a new authentication technique using message digest 5 (MD5), hashing for dynamic MANET on demand protocol (DYMO) based on reinforcement learning. In addition, we will describe an encryption technique that can be used without the need for a third party to distribute a secret key. After implementing the suggested model, results showed a remarkable enhancement in securing the path by increasing the packet delivery ratio and average throughput. On the other hand, there was an increase in end to end delay due to time spent in cryptographic operations.**

Index terms: MANET, DYMO, authentication, encryption, reinforcement.

I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a collection of mobile nodes which interacts together via wireless connections and change their location dynamically. When two nodes (Source (S) and Destination (D)) want to communicate with each other, MANET tries to create a path between the specified nodes. As shown in figure 1, this path includes many other nodes (intermediate Nodes) which transmit the message between S and D.

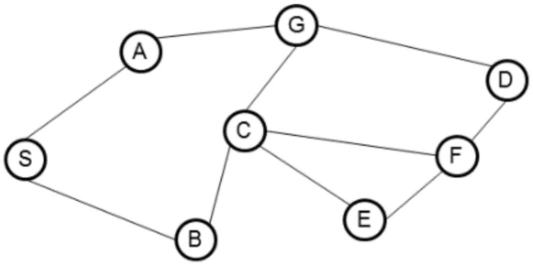


Figure1: MANET nodes

There are two main protocols used to specify the path between S and D, which are described as Table Driven and On Demand.

A. Table Driven or Proactive Protocols:

In this type of protocols, routes are created only when required [1], following is an example for this protocol.

Destination-Sequenced Distance-Vector (DSDV):
In this protocol, routing discovery process does not need much time, so it is classified as fast way to transmit packets between S and D. Routing table in DSDV updates frequently, this will consume the limited resources like battery power and the bandwidth, even when the network is inactive. If any changes happen in the network infrastructure, the routing table should change accordingly before the network rearrange. For that reason, DSDV is not considered appropriate for networks with high mobility environment or large-scale networks [2].

B. On Demand or Reactive Protocols:

In this protocol, each node creates one or more tables that include routing information to all the remaining nodes in the network. Following are examples of this protocol:

1. **Dynamic source routing (DSR):**
All types of reactive protocols have the same main features: route discovery and route maintenance. Routing discovery process: S broadcasts routing request (RREQ) to neighbouring nodes. When RREQ arrives the neighbouring node, the node retransmit it until it reaches D. D will send routing replay (RREP) message to answer the request. Any route discovery process may produce many routes to D. the longer the packet moves through the route. packet header size grows due to source routing as shown figure 2.
Route maintenance mechanism starts immediately by the middle node when an interruption occurs in the next hop that is linked to the destination node, in this case S node sends a route error (RERR) messages, after receiving RERR, S node start searching for alternative route or starts a new route discovery process [3].

978-1-7281-2337-0/19/$31.00 ©2019 IEEE
DOI 10.1109/ICPET.2019.00032

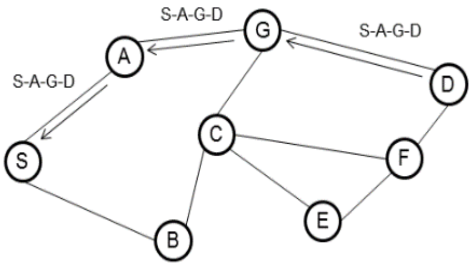


Figure 2: DSR protocol

2. **Ad hoc On-Demand Distance Vector (AODV):** AODV tries to enhance DSR by including routing tables in the nodes, as shown in figure 3, in this way data packets do not have to contain routes. AODV uses small messages defined as HELLO messages to determine local connectivity, which will shorten the time required to respond to routing requests, and activate updates if required. Sequence numbers are assigned to routes and routing table entries to take place of old cached routing entries [4].

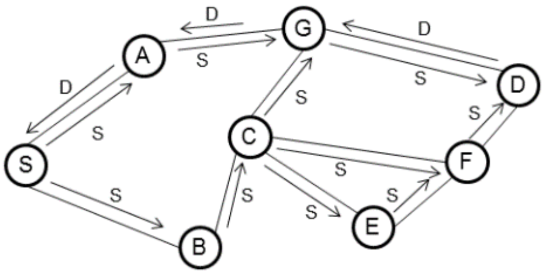


Figure 3: AODV protocol

3. **Dynamic MANET on demand (DYMO):** DYMO is a simpler version of AODV, but with path accumulation feature as shown in figure 4. Path accumulation reduces the number of RREQs and makes the route maintenance easier [5].

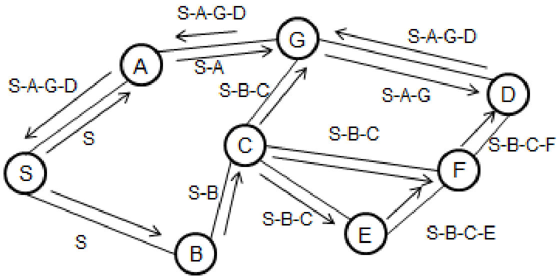


Figure 4: path accumulation in DYMO

II. PROBLEM STATEMENT

In this paper, we try to describe a way to guarantee that the message will be securely sent from S to D without being attacked by any nodes in the path. Moreover, we will explain how to recover the path between S and D in fast and efficient way if any failure occurred while transferring the data.

These problems are found in an environment that has properties such as: nodes have finite power, limited bandwidth and no centralization media [6].

III. OBJECTIVES

The main objective is to make a fast authentication between intermediate nodes and between S and D by minimizing the delay that occurs when nodes start contacting each other. Another one is securing the message between S and D by using a specific method for encryption. The last objective is to find a new way for the authentication between the nodes by using hashing algorithm.

IV. MANET SECURITY MECHANISIM

Many security mechanisms are available for MANET; these mechanisms are summarized as follows:

Certificate (C): in this mechanism, the distribution for certificates is done by nodes themselves, there is no need for trusted party to manage certificates. Nodes distribute the certificates in an independent manner. Certifications are released with determined time interval. The nodes can update the certificates before they are expired [7].

Tickets (T): Probe packets (PKT) are released with determined number of “tickets”. Many routes can be checked to determine whether they are appropriate for transmitting data or not, this checking done by tickets [8].

Digital Signatures (DS): Digital signature scheme includes: a key generation algorithm, signature algorithm and verification algorithm.

As an example for signature:

Source floods signed RREQ, then transmit it with certificates RREQ which include IP address for D.

S sends RREQ = [RREQ|| IP$_D$ || C[S]] DS$_S$

first node adds its own signature and certificates

A resends RREQ = [[RREQ|| IP$_D$ || C[S]]DS$_S$]DS$_A$ || C[A]

Each node checks the previous node signature, then substitute it with its own, then adds a reverse route to S.

B resends RREQ = [[RREQ|| IP$_D$ || C[S]]DS$_S$]DS$_B$ || C[B]

D also checks S signature[9].

Cookies session (Cookie): a cookie is a data packet, transmits from the S to D, and then transmits back to the S when D try to read this packet.

Cookies are transmitted from S to D using “Set-Cookie” headers:

Set-Cookie: NAME=VALUE; expires=DATE; path=PATH; domain=DOMAIN_NAME; secure [10].

Security Association (SA): it is aimed to create a secure communication by constructs shared security features between any two nodes in the network.

Many protocols help SA to implement several functions, here is some examples for those protocols: Authentication Headers (AH), Encapsulating Security Payloads (ESP) and Key Management Protocol.

SA features [Cryptographic algorithm || mode (Tunnel or Transport) || traffic encryption key|| parameters for the network data to be passed over the connection (Addresses) [11].

The authentication techniques that mentioned above work in different layers in OSI model. Table 1 show that layers.

Table I: Authentication techniques and OSI layers

Authentication techniques	Layer
Certificates	application
Tickets	application
Digital signatures	application
Cookies session	session
Security association	network
RAD	network

V. HASHING TECHNIQUES

Hashing are assigned values, resulted by using a specific mathematical functions. Also, it is a one-way operation to guarantee the security for data while transmitting it on the route of network. The message is designed for a determine receiver only, and the packets will be secured against tampering. We will introduce three types of Hashing:

Secure Hash Algorithm 1 (SHA-1): it is process of generation a 160-bit message digest (MD). Hence, the MD is included into a Digital Signature Algorithm (DSA) which generates/verifies the signature for the message. Due to the tiny size of MD, signing MD process will give the efficiency for the message, then the hash function will check and verify that message [12].

Secure Hash Algorithm 512 (SHA-512): it creates a digest of 512 bits from a multiple-block message. SHA-512 is a new version of SHA-1. To make SHA-512 reliable, it must have the ability to create the longest hash value that any hash function can creates, which equal to 512-bit. By using long hash value, SHA-512 will be more powerful against any attack than the function which use a small hash value. Also, SHA-512 is describe as a powerful, and high speed hash function [13].

Message digest 5 (MD5): it is a hashing algorithm specialized in processing the data in 512-bit blocks, it works in creates 16 words, each one contains 32bit. This process will produce a 128-bit message digest value. Generating MD5 is faster than SHA-1 [14].

In our model, we will use MD5 hashing because it is fast and consumes less memory. This is shown in figure 5 which describes the total bit length in the three techniques for hashing.

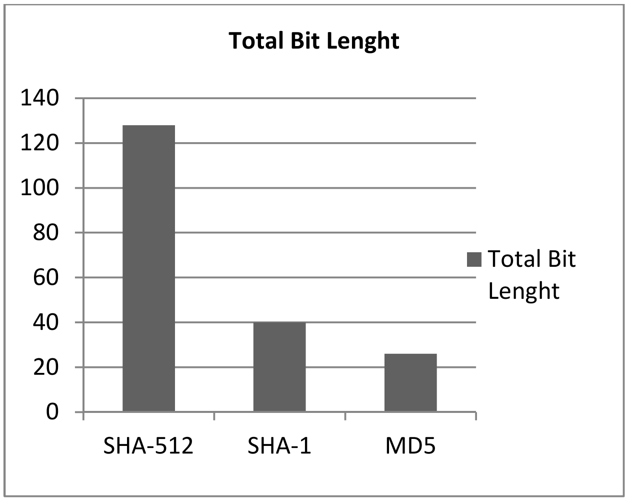


Figure 5: comparison between SHA 512, SHA1 and MD5 based on total bit length.

VI. ENCRYPTION TECHNIQUE – DIFFIE HELLMAN

Diffie-Hellman is a process to create a shared secret key between two nodes, by using a specific technique to share that key through the middle nodes without give those nodes the ability to discover it. It is good way to share a secret key between two nodes without need for third party, because the two nodes share on the key generation process [15].

VII. REINFORCEMENT LEARNING

Reinforcement protocol is a software aims to maximize the utility U(S) by taking the right action in network. When any node in network works in the right way, it will achieve some rewards. If that node takes a wrong action during transmitting the data, it will be sent to revocation list and stay there until the network test this node again in other discovered routes [16].

Here is an algorithm for reinforcement learning:

- ST– set of states of paths
- A– set of actions a (like selection of the path)
- T(st,a,st') = P(st'|st,a)– the probability of transmission from *st* to *st'* given action *a* (success in transfer data to neighbor nodes in the path)
- R(st,a)– the reward for taking action *a* in state *st* [17].

$$R(st,a) = \sum_{st'} P(st' \mid st,a) r(st,a,st')$$

$$R(st,a) = \sum_{st'} T(st,a,st') r(st,a,st')$$

$$U(st) = \max_{a}(R(st,a) + \gamma \sum_{st'} T(st,a,st') U(st'))$$

Here is the function for reinforcement of nodes in a path[18]:

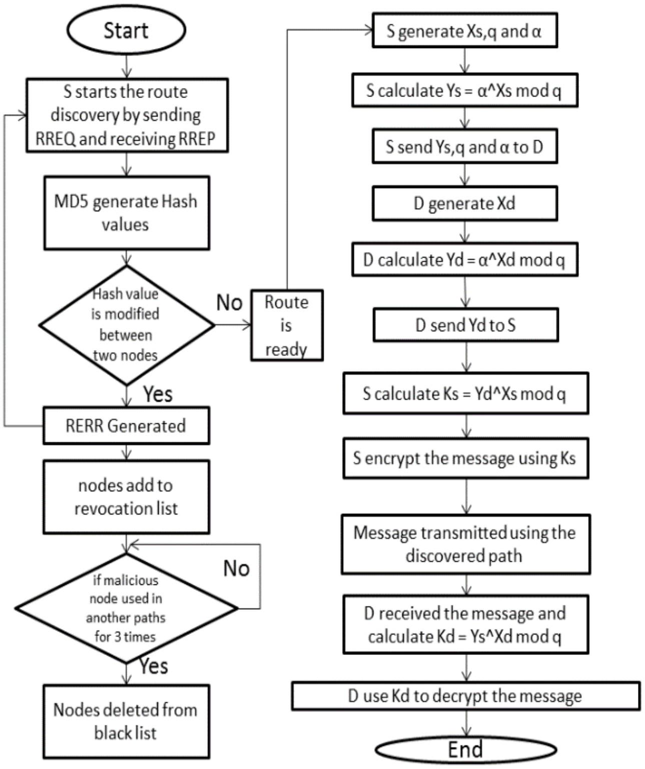
```
function NODES-REWARD (node, reward ← 1)
if node succeed to transfer data? then
    {
        reward ← reward +
                end
    }
else
{
do {
node added to revocation list
}
While (reward<3)
end
}
```

VIII. PROPOSED ALGORITHM

We propose the algorithm in the following steps:
Step 1: The source node S starts the route discovery phase by broadcasting the RREQ packet to all its neighbouring nodes.
Step 2: For authentication between nodes, MD5 algorithm will produce hash value between each intermediate nodes and between S and D.
Step 3: if there are any changes in hash value between two neighboured nodes, these two nodes will be considered as malicious and stored in the black list.
Step 4: the path discovery will continue from the node where the cut was happened, and there will be no need to restart the discovery from S.
Step 5: if any nodes from the black list used in another path for 3 times, we can treat it as normal node and delete it from the black list.
Step 6: The packets are transmitted from source to destination through the discovered path.
Step 7: Source encrypts the message using Deffie-Hellman:

a. S and D agree on p and α
b. S chooses X_S and sends $Y_S = \alpha^{Xs}$ mod q
c. D chooses X_D and sends $Y_D = \alpha^{Xd}$ mod q
d. S computes $K_S = Yd^{Xs}$ mod q
e. D computes $K_D = Yd^{Xd}$ mod q
f. Then K is the shared secret.

Step 8: Destination decrypts the message using K.
The following flow chart describes our model:



Start
S starts the route discovery by sending RREQ and receiving RREP
MD5 generate Hash values
Hash value is modified between two nodes
No
Route is ready
Yes
RERR Generated
nodes add to revocation list
if malicious node used in another paths for 3 times
No
Yes
Nodes deleted from black list
S generate Xs,q and α
S calculate Ys = α^Xs mod q
S send Ys,q and α to D
D generate Xd
D calculate Yd = α^Xd mod q
D send Yd to S
S calculate Ks = Yd^Xs mod q
S encrypt the message using Ks
Message transmitted using the discovered path
D received the message and calculate Kd = Ys^Xd mod q
D use Kd to decrypt the message
End

Figure 6: flow chart for proposed algorithm

IX. PERFORMANCE EVALUATION

In this part, we compare the protocol DYMO with the proposed protocol RAD. NS2 will be used for simulation to perform that comparison based on four points. These points are: Throughput, End to End Delay, Packet Delivery ratio and Packet Loss ratio. Table 2 shows the parameters which will be used in NS2 simulation.
These indicators analyse the performance as follows:

1. Throughput: is the amount of bits delivered per unit of time.
2. End to End Delay: is the time needed to deliver the packets from S to D.
3. Packet Delivery ratio: is the ratio of the number of packets successfully received to the number of total packets sent.
4. Packet Loss ratio: is the rate at which information is not sent through the network

Table II: NS2 simulation parameters

Simulation Tool	NS-2.35
Operating System	Ubuntu 12.04
No. of Nodes	10,20,30,40,50
Antenna model	Omni directional
Interface queue size	50 packets
Transmission range	250m
Examined protocol	DYMO, RAD
Simulation area	1100M*1100M

Figure 6 shows the throughput for DYMO and RAD. RAD has better throughput than DYMO because RAD has a guaranteed delivery for packets in unit of time. In DYMO, the rate for dropped packets is higher than RAD which makes negative effect for throughput.

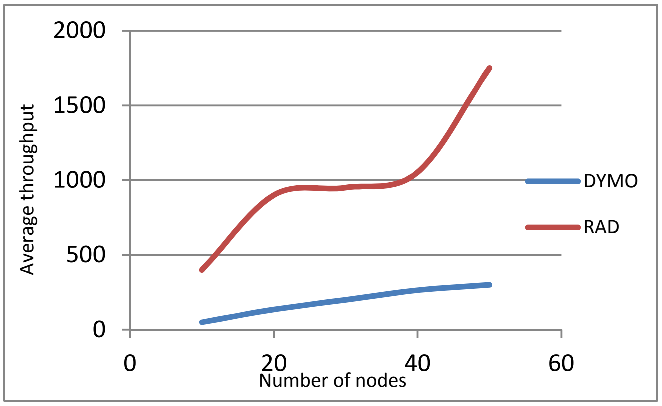


Figure 6: Average throughput

Figure 7 shows that packet delivery ratio for RAD is always better than DYMO. This happens because RAD avoids the paths which include malicious nodes.

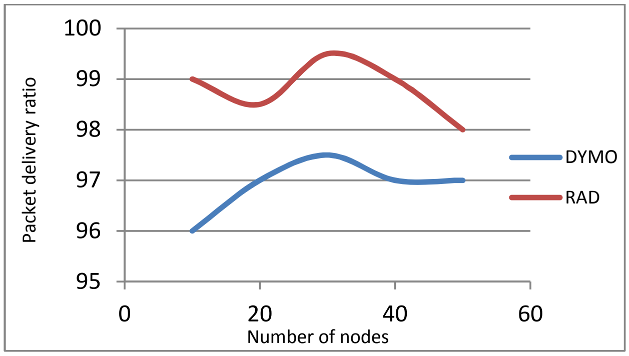


Figure 7: Packet delivery ratio

Figure 8 shows that packet loss ratio for RAD is less than DYMO. This proves that RAD ensures that packets will be transmitted in a secured path.

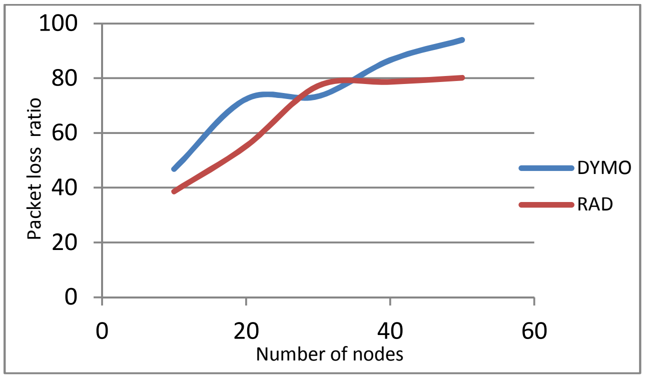


Figure 8: Packet loss ratio

Figure 9 shows that DYMO have less delay than RAD. This happens because RAD needs time to encrypt the packets and deal with hash values between nodes.

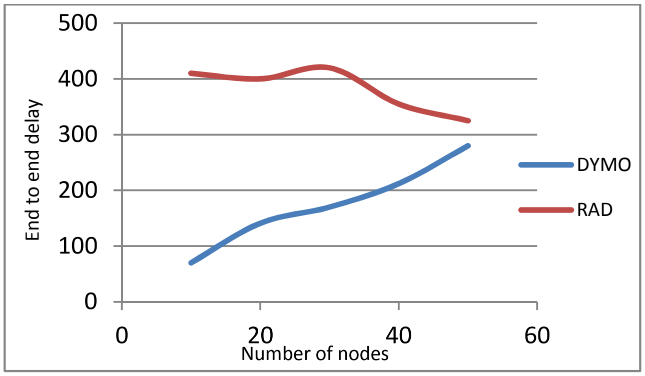


Figure 9: End to end delay

X. CONCLUSIONS

In this paper, we described a new protocol for MANET environment which is called RAD. RAD protocol is based on MD5, Diffie-Hellman and Reinforcement techniques. MD5 for hashing makes the neighboring nodes authenticated. Diffie-Hellman model distributes K_S through public key infrastructure. Reinforcement technique improves the utility of the networks through avoiding path which includes malicious nodes.

REFERENCES

1. Alslaim, M.N., H.A. Alaqel, and S.S. Zaghloul. *A comparative study of MANET routing protocols*. in *The Third International Conference on e-Technologies and Networks for Development (ICeND2014)*. 2014.
2. Khan, K.U.R., et al. *An Efficient DSDV Routing Protocol for Wireless Mobile Ad Hoc Networks and its Performance Comparison*. in *2008 Second UKSIM European Symposium on Computer Modeling and Simulation*. 2008.
3. Lili, P. *Research on performance of on- demand routing protocols in "linear structure" of Ad hoc network*. in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. 2016.
4. Gupta, A., et al. *Comparison of various routing algorithms for VANETS*. in *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*. 2016.
5. Hamamreh, R. and O. Salah, *An Intelligent Routing Protocol Based on DYMO for MANET.* International Journal of Digital Information and Wireless Communications, 2018. **8**.
6. Dhar, S., *MANET: Applications, Issues, and Challenges for the Future.* IJBDCN, 2005. **1**: p. 66-92.

7. Forne, J., et al., *Certificate status validation in mobile ad hoc networks.* IEEE Wireless Communications, 2009. **16**(1): p. 55-62.
8. Salah, S., et al., *A Model for Incident Tickets Correlation in Network Management.* Journal of Network and Systems Management, 2015. **24**.
9. Prasad, A. and K. Kaushik, *Digital Signatures*. 2019. p. 249-272.
10. Stallings, W., *Cryptography and Network Security Principles and Practices*. 2010.
11. Benin, A., S. Toledo, and E. Tromer. *Secure Association for the Internet of Things*. in *2015 International Workshop on Secure Internet of Things (SIoT)*. 2015.
12. Xiao-hui, C. and D. Jian-zhi. *Design of SHA-1 Algorithm Based on FPGA*. in *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*. 2010.
13. Gueron, S., S. Johnson, and J. Walker. *SHA-512/256*. in *2011 Eighth International Conference on Information Technology: New Generations*. 2011.
14. Yong-Xia, Z. and Z. Ge. *MD5 Research*. in *2010 Second International Conference on Multimedia and Information Technology*. 2010.
15. Nan, L. *Research on Diffie-Hellman key exchange protocol*. in *2010 2nd International Conference on Computer Engineering and Technology*. 2010.
16. Liang, X., et al., *A Deep Reinforcement Learning Network for Traffic Light Cycle Control.* IEEE Transactions on Vehicular Technology, 2019. **68**(2): p. 1243-1253.
17. Kiumarsi, B., et al., *Optimal and Autonomous Control Using Reinforcement Learning: A Survey.* IEEE Transactions on Neural Networks and Learning Systems, 2018. **29**(6): p. 2042-2062.
18. Takada, K., H. Iizuka, and M. Yamamoto, *Reinforcement Learning to Create Value and Policy Functions using Minimax Tree Search in Hex.* IEEE Transactions on Games, 2019: p. 1-1.