ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (283-296), Month: April-June 2014, Available at: www.researchpublish.com

RAACK-Reinforce Adaptive Acknowledgement A Secure Intrusion Detection System for MANETS

Ayesha Taranum[1] , Manju N[2], Tejaswini R M[3]

[1,3](PG Student) Software Engineering, Sri Jayachamarajandra College of Engineering,Mysore,Karnataka, India
[2]Assistant Professor, Dept of IS & E, Sri Jayachamarajandra College of Engineering,Mysore,Karnataka, India

Abstract: **A mobile ad hoc network (MANET) is a self-configuring infrastructure less network of mobile devices connected by wireless. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. MANET does not require a fixed network infrastructure; every single node works as both a transmitter and a receiver. Nodes communicate directly with each other when they are both within the same communication range. Otherwise, they rely on their neighbors to relay messages. The open medium and wide distribution of nodes make MANET vulnerable to malicious attackers. A new intrusion detection system named Reinforce Adaptive ACKnowledgement (RAACK) specially designed for MANETs. By the adoption of MRA scheme, RAACK is capable of detecting malicious nodes despite the existence of false misbehavior report. In this paper, we propose and implement a new intrusion-detection system named Reinforce Adaptive ACKnowledgment (RAACK) specially designed for MANETs. Compared to contemporary approaches, RAACK demonstrates higher malicious-behavior-detection rates.**

Keywords: **Elliptic Curve Digital Signature Algorithm (ECDSA), Elliptic Curve Cryptography (ECC), MANET.**

I. INTRODUCTION

The latest trend in wireless networks is towards *pervasive and ubiquitous computing* - catering to both nomadic and fixed users, anytime and anywhere. In such a network, a set of mobile nodes are connected to a fixed wired backbone. WLANs have a short range and are usually deployed in places such universities, companies, cafeterias. Mobile Ad hoc NETwork (MANET) is a collection of mobile nodes equipped with both a wireless transmitter and a receiver that communicate with each other via bidirectional wireless links either directly or indirectly. Industrial remote access and control via wireless networks are becoming more and more popular these days [35]. One of the major advantages of wireless networks is its ability to allow data communication between different parties and still maintain their mobility. However, this communication is limited to the range of transmitters. This means that two nodes cannot communicate with each other when the distance between the two nodes is beyond the communication range of their own. MANET solves this problem by allowing intermediate parties to relay data transmissions. This is achieved by dividing MANET into two types of networks, namely, single-hop and multihop. In a single-hop network, all nodes within the same radio range communicate directly with each other. On the other hand, in a multihop network, nodes rely on other intermediate nodes to transmit if the destination node is out of their radio range. In contrary to the traditional wireless network, MANET has a decentralized network infrastructure. MANET does not require a fixed infrastructure; thus, all nodes are free to move randomly [10], [27], [29]. MANET is capable of creating a self-configuring and self-maintaining network without the help of a centralized infrastructure, which is often infeasible in critical mission applications like military conflict or emergency recovery. Minimal configuration and quick deployment make MANET ready to be used in emergency circumstances where an infrastructure is unavailable or unfeasible to install in scenarios like natural or human-induced disasters, military conflicts, and medical emergency situations [19], [30].

II. BACKGROUND

A. IDS in MANETs

As discussed before, due to the limitations of most MANET routing protocols, nodes in MANETs assume that other nodes always cooperate with each other to relay data. This assumption leaves the attackers with the opportunities to achieve significant impact on the network with just one or two compromised nodes. To address this problem, an IDS should be added to enhance the security level of MANETs. If MANET can detect the attackers as soon as they enter the network, we will be able to completely eliminate the potential damages caused by compromised nodes at the first time. IDSs usually act as the second layer in MANETs, and they are a great complement to existing proactive approaches [27]. Anantvalee and Wu [4] presented a very thorough survey on contemporary IDSs in MANETs. In this section, we mainly describe three existing approaches, namely, Watchdog [17], TWOACK [15], and Adaptive ACKnowledgment (AACK) [25]. *1) Watchdog:* Marti *et al.* [17] proposed a scheme named Watchdog that aims to improve the throughput of network with the presence of malicious nodes. In fact, the Watchdog scheme is consisted of two parts, namely, Watchdog and Pathrater.Watchdog serves as an IDS for MANETs. It is responsible for detecting malicious node misbehaviors in the network. Watchdog detects malicious misbehaviors by promiscuously listening to its next hop's transmission. If a Watchdog node overhears that its next node fails to forward the packet within a certain period of time, it increases its failure counter. Whenever a node's failure counter exceeds a predefined threshold, the Watchdog node reports it as misbehaving. In this case, the Pathrater cooperates with the routing protocols to avoid the reported nodes in future transmission. Many following research studies and implementations have proved that the Watchdog scheme is efficient. Furthermore, compared to some other schemes, Watchdog is capable of detecting malicious nodes rather than links. These advantages have made the Watchdog scheme a popular choice in the field. Many MANET IDSs are either based on or developed as an improvement to the Watchdog scheme [15], [20], [21], [25]. Nevertheless, as pointed out by Marti *et al.* [17], the Watchdog scheme fails to detect malicious misbehaviors with the presence of the following: 1) ambiguous collisions; 2) receiver collisions; 3) limited transmission power; 4) false misbehavior report; 5) collusion; and 6) partial dropping. We discuss these weaknesses with further detail in Section III.*2) TWOACK:* With respect to the six weaknesses of the Watchdog scheme, many researchers proposed new approaches to solve these issues. TWOACK proposed by Liu *et al.* [16] is one of the most important approaches among them. On

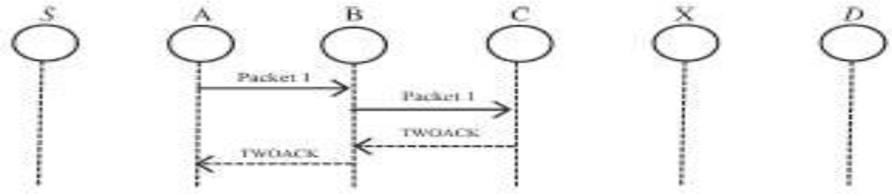


Fig. 1. TWOACK scheme

Each node is required to send back an acknowledgment packet to the node that is two hops away from it. The contrary to many other schemes, TWOACK is neither an enhancement nor a Watchdog-based scheme. Aiming to resolve the receiver collision and limited transmission power problems of Watchdog, TWOACK detects misbehaving links by acknowledging every data packet transmitted over every three consecutive nodes along the path from the source to the destination. Upon retrieval of a packet, each node along the route is required to send back an acknowledgment packet to the node that is two hops away from it down the route. TWOACK is required to work on routing protocols such as Dynamic Source Routing (DSR) [11]. The working process of TWOACK is shown in Fig. 1: Node A first forwards Packet 1 to node B, and then, node B forwards Packet 1 to node C. When node C receives Packet 1, as it is two hops away from node A, node C is obliged to generate a TWOACK packet, which contains reverse route from node A to node C, and sends it back to node A. The retrieval of this TWOACK packet at node A indicates that the transmission of Packet 1 from node A to node C is successful. Otherwise, if this TWOACK packet is not received in a predefined time period, both nodes B and C are reported malicious. The same process applies to every three consecutive nodes along the rest of the route. The TWOACK scheme successfully solves the receiver collision and limited transmission power problems posed by Watchdog. However, the acknowledgment process required in every packet transmission process added a significant amount of unwanted network overhead. Due to the limited battery power nature of MANETs, such redundant transmission process can easily degrade the life span of the entire network. However, many research studies are working in energy harvesting to deal with this problem [25], [28], [29]. *3) AACK:* Based on TWOACK, Sheltami *et al.* [25] proposed a new scheme called AACK. Similar to TWOACK, AACK is an acknowledgment-based network layer scheme which can be considered as a combination of a scheme called TACK (identical to TWOACK) and an end-to-end acknowledgment scheme called ACKnowledge (ACK). Compared to TWOACK, AACK significantly reduced network overhead while still capable of maintaining or even surpassing the same network throughput. The end-to-end acknowledgment scheme in ACK is shown in Fig. 2. In the ACK scheme shown in Fig. 2, the source node S sends out Packet 1 without any overhead except 2 b of flag indicating the packet type. All the intermediate nodes simply forward this packet. When the destination node D receives Packet 1, it is required to send back an ACK acknowledgment packet to the source node S along the reverse order of the

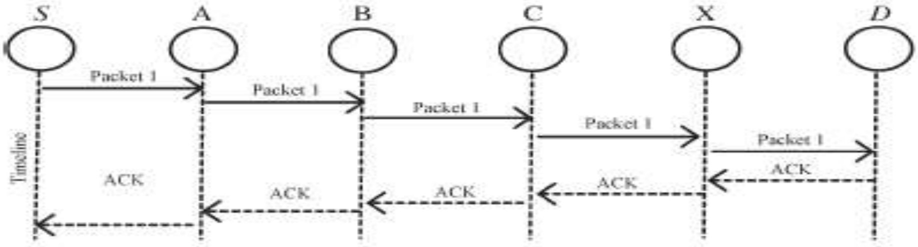


Fig. 2. ACK scheme

The destination node is required to send acknowledgment packets to the source node. same route. Within a predefined time period, if the source node S receives this ACK acknowledgment packet, then the packet transmission from node S to node D is successful. Otherwise, the source node S will switch to TACK scheme by sending out a TACK packet. The concept of adopting a hybrid scheme in AACK greatly reduces the network overhead, but both TWOACK and AACK still suffer from the problem that they fail to detect malicious nodes with the presence of false misbehavior report and forged acknowledgment packets. In fact, many of the existing IDSs in MANETs adopt an acknowledgment-based scheme, including TWOACK and AACK. The functions of such detection schemes all largely depend on the acknowledgment packets. Hence, it is crucial to guarantee that the acknowledgment packets are valid and authentic. To address this concern, we adopt a digital signature in our proposed scheme named Enhanced AACK (RAACK).

B. Digital Signature

In public key cryptography each person has a pair of keys: a public key and a private key. These are typically numbers that are chosen to have a specific mathematical relationship. In RSA, the public key is a large number that is a product of two primes, plus a smaller number. The private key is a related number. In ECC, the public key is an equation for an elliptic curve and a point that lies on that curve. The private key is a number. See our previous blog post on elliptic curve cryptography for more details. The private key can be used to create a digital signature for any piece of data using a digital signature algorithm. This typically involves taking a cryptographic hash of the data and operating on it mathematically using the private key. Anyone with the public key can check that this signature was created using the private key and the appropriate signature validation algorithm. A digital signature is a powerful tool because it allows you to publicly vouch for any message. A website certificate usually contains two things: Identity information: Typically who owns the certificate and which domains the certificate is valid for. A public key: The public half of a key pair, the site owner controls and keeps secret the associated private key. The certificate is digitally signed by a trusted certificate authority who validates the identity of the site owner. Since the introduction of SSL by Netscape in 1994, certificates for web sites have typically used a public/private key pair based on the RSA algorithm. As the SSL specification evolved into TLS, supports for different public key algorithms were added. One of the supported algorithms is ECDSA which is based on elliptic curves. Despite the number of options available in TLS, almost all certificates used on the web today are RSA-based. Web sites have been slow to adopt new algorithms because they want to maintain support for legacy browsers that don't support the new algorithms. Even as late as 2012, out of 13 million TLS certificates found in a scan of the internet, fewer than 50 use an ECDSA key pair. Elliptic Curve Digital Signature Algorithm is implemented over elliptic curve P-192 as mandated by ANSI X9.62 in C language. It contains necessary modules for domain parameters generation, key generation, signature generation, and signature verification over the elliptic curve. ECDSA has three phases, key generation, signature generation, and signature verification. In this research work, we implemented ECDSA in our proposed RAACK scheme. The main purpose of this implementation is to compare their performances in MANETs

1) ECDSA Key Generation:

An entity A’s key pair is associated with a particular set of EC domain parameters D= (q, FR, a, b, G, n, h). E is an elliptic curve defined over Fq , and P is a point of prime order n in E(Fq), q is a prime. Each entity A does the following: 1. Select a random integer d in the interval [1, n- 1]. 2. Compute Q = dP. 3. A’s public key is Q, A’s private key is d.

2) ECDSA Signature Generation:.

To sign a message m, an entity A with domain parameters D= (q, FR, a, b, G, n, h) does the following: 1. Select a random or pseudorandom integer k in the interval [1, n-1]. 2. Compute kP =x1, y1 and r= x1 mod n (where x1 is regarded as an integer between 0 and q-1). If r= 0 then go back to step 1. 3. Compute k-1mod n. 4. Compute s= k-1 {h (m) + dr} mod n, where h is the Secure Hash Algorithm (SHA-1). If s = 0, then go back to step 1. 5. The signature for the message m is the pair of integers (r, s).

3) ECDSA Signature Verification:

To verify A’s signature (r, s) on m, B obtains an authenticated copy of A’s domain parameters D = (q, FR, a, b, G, n, h) and public key Q and do the following 1. Verify that r and s are integers in the interval [1, n-1]. 2. Compute w = s-1mod n and h (m) 3. Compute u1 = h(m)w mod n and u2 = rw mod n. 4. Compute u1P + u2Q =(x0, y0) and v= x0 mod n. 5. Accept the signature if and only if v = r

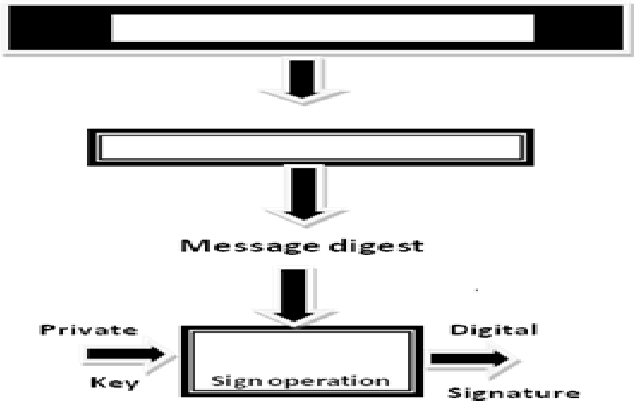


Figure 3. Signature Generation

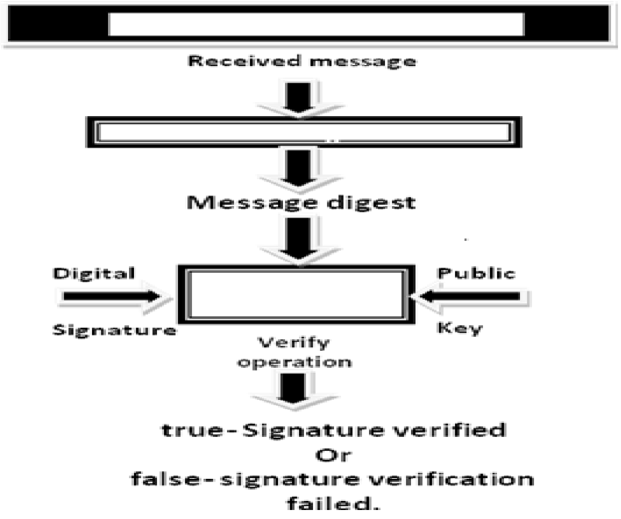


Figure 4. Signature Verification

Results

The following results are brought to highlight for given set of values. The SHA-1 result are shown along with the private and public set of keys

SHA–1

Input: “a”

SHA Output: 86f7e437faa5a7fce15d1ddcb9eaeaea377667b8

Input: “ABC”

SHA Output: 3c01bdbb26f358bab27f267924aa2c9a03fcfdb8

Key Pair Generation:

198 bit random private key and corresponding public key:

Private A=**3410708343957475413710496549104959138812316708511486831983**

Public x of A=**3089182225850909019933101519334356466906901301271156815371**

Public y of A=**2934312592567055080539106109257350191706192298057173813254**

Private A=**97847545072694784411473994099387459926335654578030561509614891**

Public x of A=**5794350039132556514670158969918976743409250716115312636030**

Public y of A=**1009024622477364832125741509919741456473929964192222324391**

Further for a given input file containing text had been taken and signature is generated and then verified by the values of r and s.

Signature Generation:

Input file="abcd"

Private: 0xd43fb7ff56a7486859d87f785db45b043129f6468ccff42d0001

Signature:

r=0xb8d06fa44816c92b8b26f797e5f3cc07984d8b7f7e49a339

s=0xd74f17a1e19139d77558c6b2d16dcb1f4bb31da2ded25733

Proof of verification

If a signature (r, s) on a message m was indeed generated by A, then s = k -1 (h (m)+dr) mod n. Rearranging gives k s-1 (e+dr) ≡s-1e + s-1 rd ≡we +wrd ≡ u1+u2d (mod n).Thus u1G +u2Q = (u1 +u2d) G = kG and so v=r as required.

4) COMPARISON WITH RSA and DSA

In all cryptography systems discussed so far, there is a comparative difficulty of doing two types of operations-a forward operation which must be tractable and an inverse operation which must be intractable. The degree of difference between the difficulties of these operations depends on the size of the key pairs. The inverse operation increases exponentially whereas the forward operation increases linearly as the key size increases as in Figure 6. Increase in key length give rise to complexity issues in both operations. Thus ECC is preferred as it provides same level security at 160 bit key length as of 1024 bit key length in RSA.

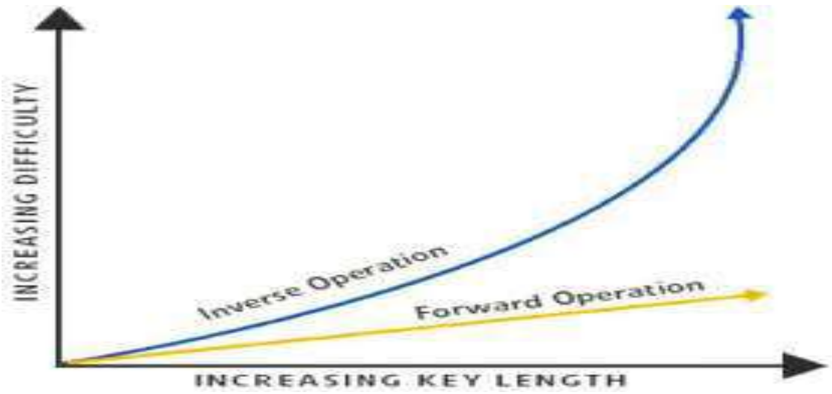


Figure 5. Difficulty of forward, inverse operation against key length

Table I show the comparison of ECC with RSA, DSA, and DH in terms of key length and time to break on machine running 1 MIPS.

Table I. Key comparison of Symmetric

Symmetric	**RSA/DSA/DH**	**ECC**	**Time to break in MIPS years**
80	1024	160	10_{12}
112	2048	224	10_{24}
128	3072	256	10_{28}
192	7680	384	10_{47}

4.1) Comparison of ECC with RSA

1. RSA takes sub-exponential time and ECC takes full exponential time. For example, RSA with key size of 1024 bits takes 3x1011 MIP years with best known attack where as ECC with 160 bit key size takes 9.6x 10^11 MIP years. **2.** ECC offers same level of security with smaller key sizes **3.** DATA size for RSA is smaller than ECC.**4.** Encrypted message is a function of key size and data size for both RSA and ECC. ECC key size is relatively smaller than RSA key size, thus encrypted message in ECC is smaller. **5.** Computational power is smaller for ECC.

4.2) Comparison of ECDSA with DSA

1. Both algorithms are based on the ElGamal signature scheme and use the same signing equation: s = k-1{h (m) + dr} mod n. **2.** In both algorithms, the values that are relatively difficult to generate are the system parameters (p, q and g for the DSA; E, P and n for the ECDSA). **3.** In their current version, both DSA and ECDSA use the SHA-1 as the sole cryptographic hash function. **4.** The private key d and the per-signature value k in ECDSA are defined to be statistically unique and unpredictable rather than merely random as in DSA.

4.3) Advantages of ECC

Thus, the ECC offered remarkable advantages over other cryptographic system. **1.** It provides greater security for a given key size. **2.** It provides effective and compact implementations for cryptographic operations requiring smaller chips. **3.** Due to smaller chips less heat generation and less power consumption.**4.** It is mostly suitable for machines having low bandwidth, low computing power, less memory. **5.** It has easier hardware implementations. So far no drawback of ECC had been reported. with elliptic curve cryptography in general, the bit size of the public key believed to be needed for

ECDSA is about twice the size of the security level, in bits. By comparison, at a security level of 80 bits (meaning an attacker requires the equivalent of about 2^{80} operations to find the private key) the size of a DSA public key constructor n = p x q is at least 1024 bits, but the public key itself can be ~ 17 bits of course that means a correspondingly larger private key, whereas the size of an ECDSA public key would be 160 bits. On the other hand, the signature size is the same for both DSA and ECDSA: $4t$bits, where $t$is the security level measured in bits, that is, about 320 bits for a security level of 80 bits.

III. PROBLEM DEFINITION

Our proposed approach RAACK is designed to tackle three of the six weaknesses of Watchdog scheme, namely, false misbehavior, alternate Path, and receiver collision. In this section, we discuss these three weaknesses in detail.

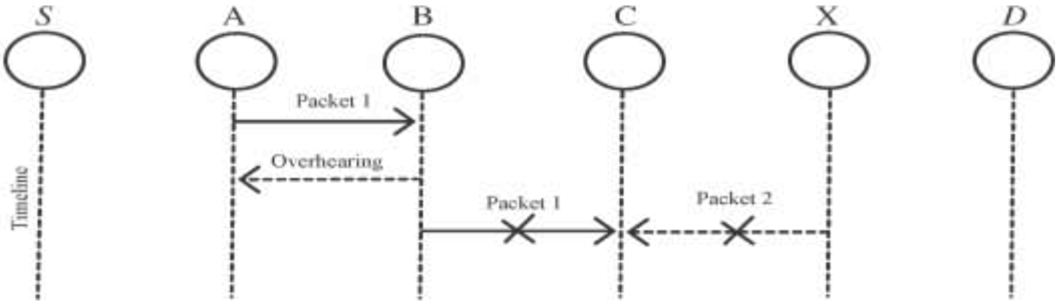


Fig. 6 Receiver collisions

Node B limits its transmission power so that the packet transmission can be overheard by node A but too weak to reach node C.

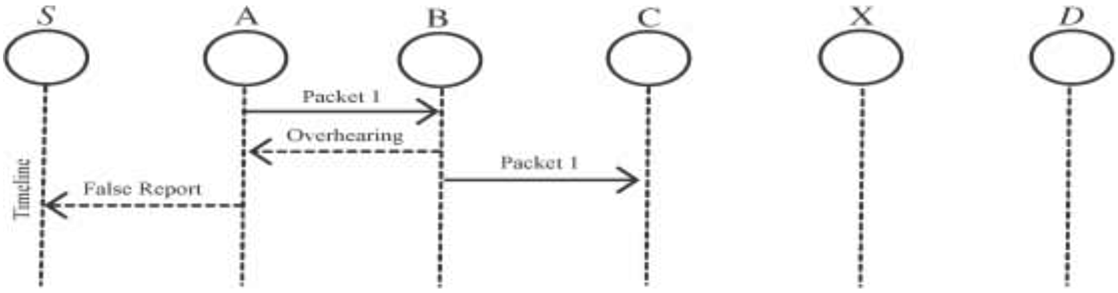


Fig.7. False misbehavior report

Node A sends back a misbehavior report even though node B forwarded the packet to node C.

In a typical example of receiver collisions, shown in Fig. 6, after node A sends Packet 1 to node B, it tries to overhear if node B forwarded this packet to node C; meanwhile, node X is forwarding Packet 2 to node C. In such case, node A overhears that node B has successfully forwarded Packet 1 to node C but failed to detect that node C did not receive this packet due to a collision between Packet 1 and Packet 2 at node C. For false misbehavior report, although node A successfully overheard that node B forwarded Packet 1 to node C, node A still reported node B as misbehaving, as shown in Fig. 6. Due to the open medium and remote distribution of typical MANETs, attackers can easily capture and compromise one or two nodes to achieve this false misbehavior report attack.

As discussed in previous sections, TWOACK and AACK solve two of these three weaknesses, namely, receiver collision and limited transmission power. However, both of them are vulnerable to the false misbehavior attack. In this research work, our goal is to propose new IDS specially designed for MANETs, which solves not only receiver collision but also the false misbehavior problem.

Furthermore, we extend our research to adopt a digital sig- nature scheme during the packet transmission process. As in all acknowledgment-based IDSs, it is vital to ensure the integrity and authenticity of all acknowledgment packets.

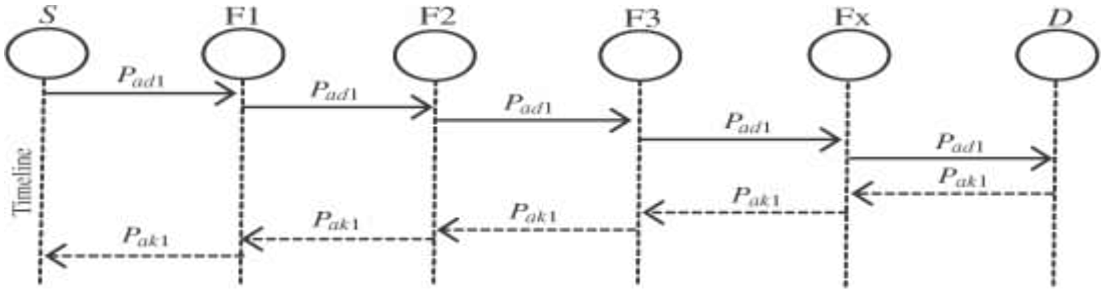


Fig.8. System control flow: This figure shows the system flow of how the RAACK scheme works.

IV. SCHEME DESCRIPTION

In this section, we describe our proposed RAACK scheme in detail. The approach described in this research paper is based on our previous work [12], where the backbone of RAACK was proposed and evaluated through implementation. In this paper, we extend it with the introduction of digital signature to prevent the attacker from forging acknowledgment packets.

RAACK is consisted of three major parts, namely, ACK, secure ACK (S-ACK), and misbehavior report authentication (MRA). In order to distinguish different packet types in different schemes, Fig.14 (shown later) presents a flowchart describing the RAACK scheme. Please note that, in our proposed scheme, we assume that the link between each node in the network is bidirectional. Furthermore, for each communication process, both the source node and the destination node are not malicious. Unless specified, all acknowledgment packets described in this research are required to be digitally signed by its sender and verified by its receiver.

A. ACK

ACK is basically an end-to-end acknowledgment scheme. It acts as a part of the hybrid scheme in RAACK, aiming to reduce network overhead when no network misbehavior is detected. In Fig. 8, in ACK mode, node S first sends out an ACK data packet P_{ad1} to the destination node D. If all the intermediate nodes along the route between nodes S and D are cooperative and node D successfully receives P_{ad1} , node D is required to send back an ACK acknowledgment packet P_{ak1} along the same route but in a reverse order. Within a predefined time period, if node S receives P_{ak1}, then the packet transmission from node S to node D is successful. Otherwise, node S will switch to S-ACK mode by sending out an S-ACK data packet to detect the misbehaving nodes in the route.

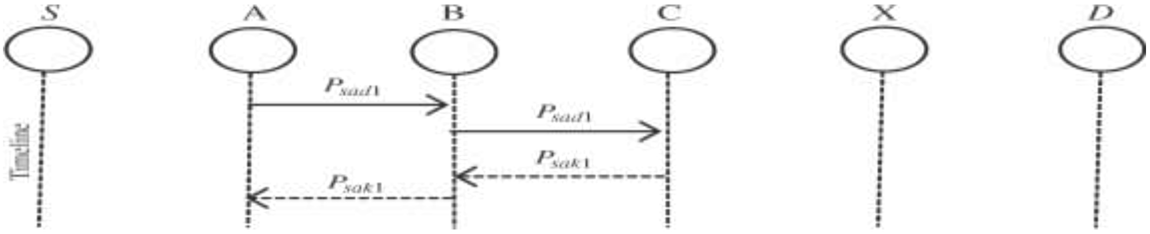


Fig .9. ACK scheme: The destination node is required to send back an acknowledgment packet to the source node when it receives a new packet.

B. S-ACK

The S-ACK scheme is an improved version of the TWOACK scheme proposed by Liu *et al.* [16]. The principle is to let every three consecutive nodes work in a group to detect misbehaving nodes. For every three consecutive nodes in the route, the third node is required to send an S-ACK acknowledgment packet to the first node. The intention of introducing S-ACK mode is to detect misbehaving nodes in the presence of receiver collision or limited transmission power.

As shown in Fig. 9, in S-ACK mode, the three consecutive nodes (i.e., F1, F2, and F3) work in a group to detect misbehaving nodes in the network. Node F1 first sends out S-ACK data packet P_{sad1} to node F2. Then, node F2 forwards this packet to node F3. When node F3 receives P_{sad1}, as it is the third node in this three-node group, node F3 is required to send back an S-ACK acknowledgment packet P_{sak1} to node F2. Node F2 forwards P_{sak1} back to node F1. If node F1 does not receive this acknowledgment packet within a predefined time period, both nodes F2 and F3 are reported as malicious. Moreover, a misbehavior report will be generated by node F1 and sent to the source node S.

Nevertheless, unlike the TWOACK scheme, where the source node immediately trusts the misbehavior report, RAACK requires the source node to switch to MRA mode and confirm this misbehavior report. This is a vital step to detect false misbehavior report in our proposed scheme.

C. MRA

The MRA scheme is designed to resolve the weakness of Watchdog when it fails to detect misbehaving nodes with the presence of false misbehavior report. The false misbehavior report can be generated by malicious attackers to falsely report innocent nodes as malicious. This attack can be lethal to the entire network when the attackers break down sufficient nodes and thus cause a network division. The core of MRA scheme is to authenticate whether the destination node has received the reported missing packet through a different route.

To initiate the MRA mode, the source node first searches its local knowledge base and seeks for an alternative route to the destination node. If there is no other that exists, the source node starts a DSR routing request to find another route. Due to the nature of MANETs, it is common to find out multiple routes between two nodes.

By adopting an alternative route to the destination node, we circumvent the misbehavior reporter node. When the destination node receives an MRA packet, it searches its local knowledge base and compares if the reported packet was received. If it is already received, then it is safe to conclude that this is a false misbehavior report and whoever generated this report is marked as malicious. Otherwise, the misbehavior report is trusted and accepted.

By the adoption of MRA scheme, RAACK is capable of detecting malicious nodes despite the existence of false misbehavior report.

D. Digital Signature

RAACK is an acknowledgment-based IDS. All three parts of RAACK, namely, ACK, S-ACK, and MRA, are acknowledgment-based detection schemes. They all rely on acknowledgment packets to detect misbehaviors in the network. Thus, it is extremely important to ensure that all acknowledgment packets in RAACK are authentic and untainted. Otherwise, if the attackers are smart enough to forge acknowledgment packets, all of the three schemes will be vulnerable.

We incorporated digital signature in our proposed scheme. In order to ensure the integrity of the IDS, RAACK requires all acknowledgment packets to be digitally signed before they are sent out and verified until they are accepted. However, we fully understand the extra resources that are required with the introduction of digital signature in MANETs. To address this concern, we implemented ECDSA digital signature schemes in our proposed approach. The goal is to find the most optimal solution for using digital signature in MANETs.

V. PERFORMANCE EVALUATION

In this section, we concentrate on describing our simulation environment and methodology as well as comparing performances through simulation result comparison with Watchdog, TWOACK, and RAACK schemes.

A. Simulation Methodologies

The performance of RAACK under different types of attacks, we propose three scenario settings to simulate different types of misbehaviors or attacks.

Scenario 1: Basic packet- dropping attack is simulated. Malicious nodes simply drop all the packets that they receive. The purpose of this scenario is to test the performance of IDSs against two weaknesses of Watchdog, namely, receiver collision and limited transmission power.

Scenario 2: To test IDSs' performances against false misbehavior report. In this case, malicious nodes always drop the packets that they receive and send back a false misbehavior report whenever it is possible.

Scenario 3: This scenario is used to test the IDSs' performances when the attackers are smart enough to forge acknowledgment packets and claiming positive result while, in fact, it is negative. As Watchdog is not an acknowledgment-based scheme, it is not eligible for this scenario setting.

In order to measure and compare the performances of our proposed scheme, we continue to adopt the following two performance metrics [13].

1) **Packet delivery ratio (PDR):** PDR defines the ratio of the number of packets received by the destination node to the number of packets sent by the source node.

2) **Routing overhead (RO):** RO defines the ratio of the amount of routing-related transmissions [Route REQuest (RREQ), Route REPly (RREP), Route ERRor (RERR), ACK, S-ACK, and MRA].

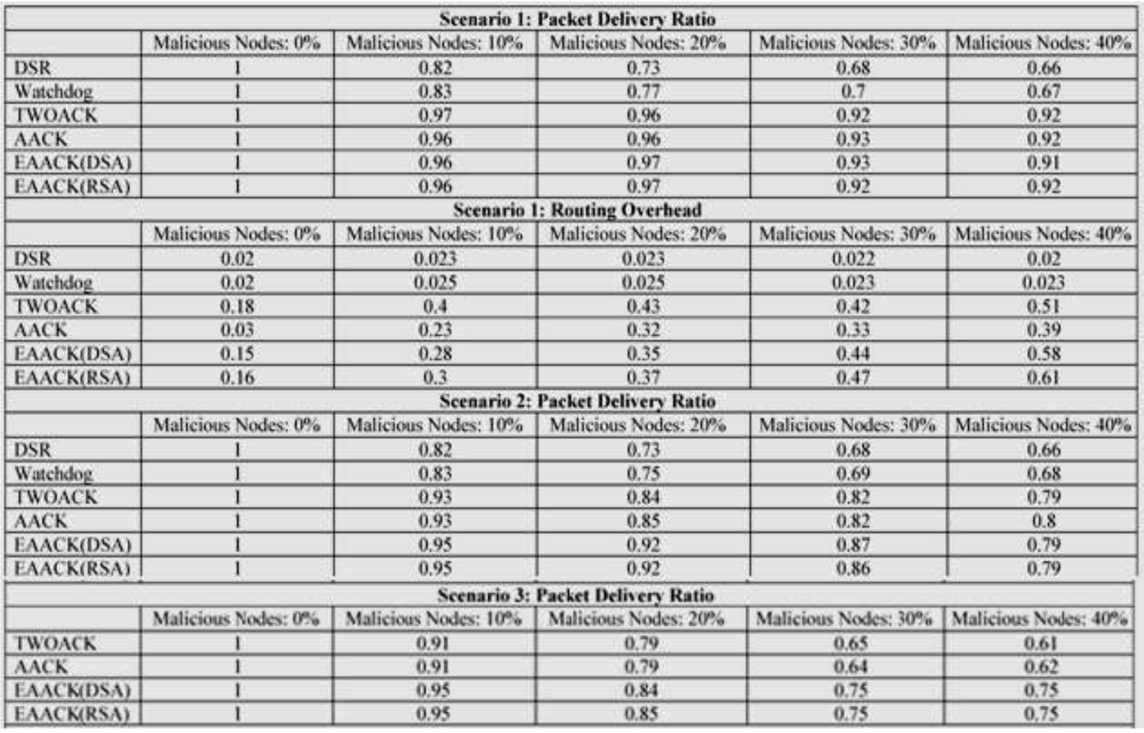
During the simulation, the source route broadcasts an RREQ message to all the neighbors within its communication range. Upon receiving this RREQ message, each neighbor appends their addresses to the message and broadcasts this new message to their neighbors. If any node receives the same RREQ message more than once, it ignores it. If a failed node is detected, which generally indicates a broken link in flat routing protocols like DSR, a RERR message is sent to the source node. When the RREQ message arrives to its final destination node, the destination node initiates an RREP message and sends this message back to the source node by reversing the route in the RREQ message.

Comparing DSA and ECDSA. Conceptually, the ECDSA is simply obtain from the DSA by replacing the subgroup of order q of (Z/pZ)× generated by g with the subgroup of points on an elliptic curve that are generated by G. The only significant difference between ECDSA and DSA is in the generation of r. The DSA does this by taking the random element X = gk mod p and reducing it modulo q, thus obtaining an integer in the interval [1,q−1]. The ECDSA generates r in the interval [1,n−1] by taking the x-coordinate of the random point kG and reducing it modulo n.

B. Performance Evaluation

To provide readers with a better insight on our simulation results, detailed simulation data are presented in Table II.

Table II



Scenario 1: Packet Delivery Ratio					
	Malicious Nodes: 0%	Malicious Nodes: 10%	Malicious Nodes: 20%	Malicious Nodes: 30%	Malicious Nodes: 40%
DSR	1	0.82	0.73	0.68	0.66
Watchdog	1	0.83	0.77	0.7	0.67
TWOACK	1	0.97	0.96	0.92	0.92
AACK	1	0.96	0.96	0.93	0.92
EAACK(DSA)	1	0.96	0.97	0.93	0.91
EAACK(RSA)	1	0.96	0.97	0.92	0.92
Scenario 1: Routing Overhead					
	Malicious Nodes: 0%	Malicious Nodes: 10%	Malicious Nodes: 20%	Malicious Nodes: 30%	Malicious Nodes: 40%
DSR	0.02	0.023	0.023	0.022	0.02
Watchdog	0.02	0.025	0.025	0.023	0.023
TWOACK	0.18	0.4	0.43	0.42	0.51
AACK	0.03	0.23	0.32	0.33	0.39
EAACK(DSA)	0.15	0.28	0.35	0.44	0.58
EAACK(RSA)	0.16	0.3	0.37	0.47	0.61
Scenario 2: Packet Delivery Ratio					
	Malicious Nodes: 0%	Malicious Nodes: 10%	Malicious Nodes: 20%	Malicious Nodes: 30%	Malicious Nodes: 40%
DSR	1	0.82	0.73	0.68	0.66
Watchdog	1	0.83	0.75	0.69	0.68
TWOACK	1	0.93	0.84	0.82	0.79
AACK	1	0.93	0.85	0.82	0.8
EAACK(DSA)	1	0.95	0.92	0.87	0.79
EAACK(RSA)	1	0.95	0.92	0.86	0.79
Scenario 3: Packet Delivery Ratio					
	Malicious Nodes: 0%	Malicious Nodes: 10%	Malicious Nodes: 20%	Malicious Nodes: 30%	Malicious Nodes: 40%
TWOACK	1	0.91	0.79	0.65	0.61
AACK	1	0.91	0.79	0.64	0.62
EAACK(DSA)	1	0.95	0.84	0.75	0.75
EAACK(RSA)	1	0.95	0.85	0.75	0.75

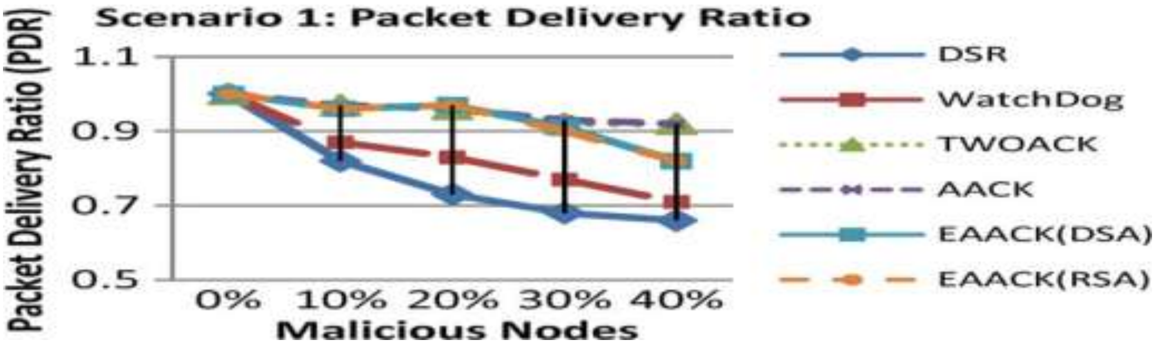


Fig. 10. Simulation results for scenario 1—PDR.

1) Simulation Results—Scenario 1: In scenario 1, malicious nodes drop all the packets that pass through it. Fig. 10 shows the simulation results that are based on PDR.

In Fig. 10, we observe that all acknowledgment-based IDSs perform better than the Watchdog scheme. Our proposed scheme RAACK surpassed Watchdog's performance by 21%

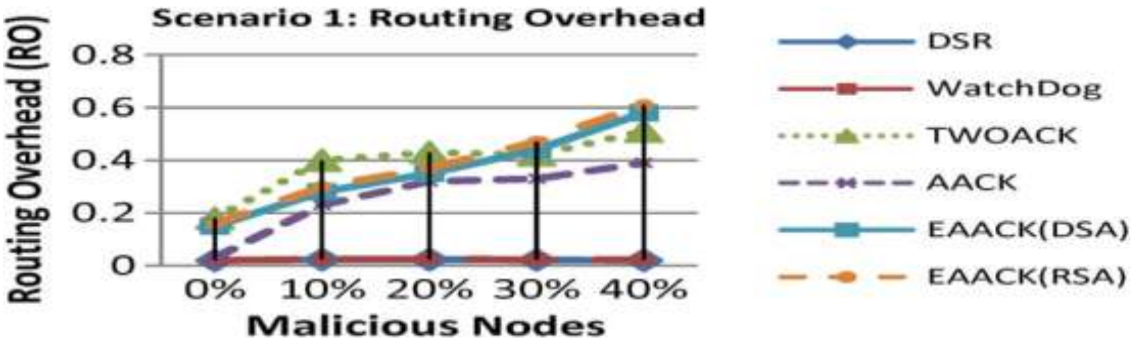


Fig. 11. Simulation results for scenario 1—RO.

When there are 20% of malicious nodes in the network. From the results, we conclude that acknowledgment based schemes, including TWOACK, AACK, and RAACK, are able to detect misbehaviors with the presence of receiver collision and limited transmission power. However, when the number of malicious nodes reaches 40%, our proposed scheme RAACK's performance is lower than those of TWOACK and AACK. We generalize it as a result of the introduction of MRA scheme, when it takes too long to receive an MRA acknowledgment from the destination node that the waiting time exceeds the predefined threshold.

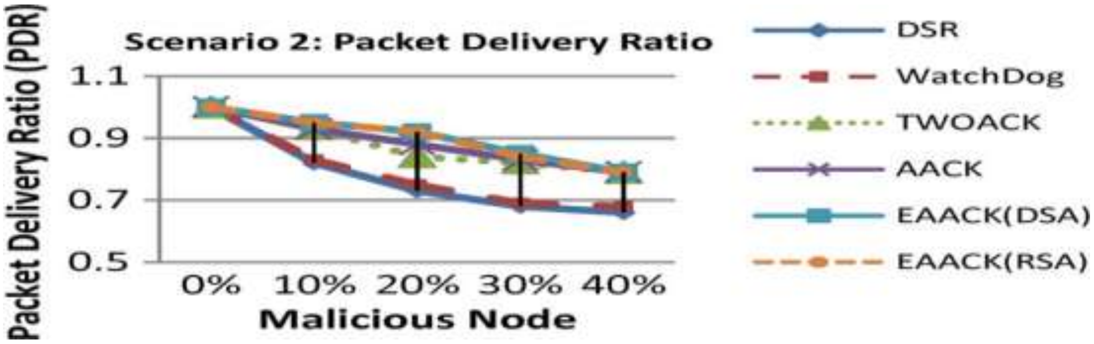


Fig. 12. Simulation results for scenario 2—PDR.

2) Simulation Results—Scenario 2: we set all malicious nodes to send out false misbehavior report to the source node whenever it is possible. This scenario setting is designed to test the IDS's performance under the false misbehavior report. Fig. 12 shows the achieved simulation results based on PDR. When malicious nodes are 10%, RAACK performs 2% better than AACK and TWOACK. When the ma- licious nodes are at 20% and 30%, RAACK outperforms all the other schemes and maintains the PDR to over 90%. We believe that the introduction of MRA scheme mainly contributes to this performance. RAACK is the only scheme that is capable of detecting false misbehavior report.

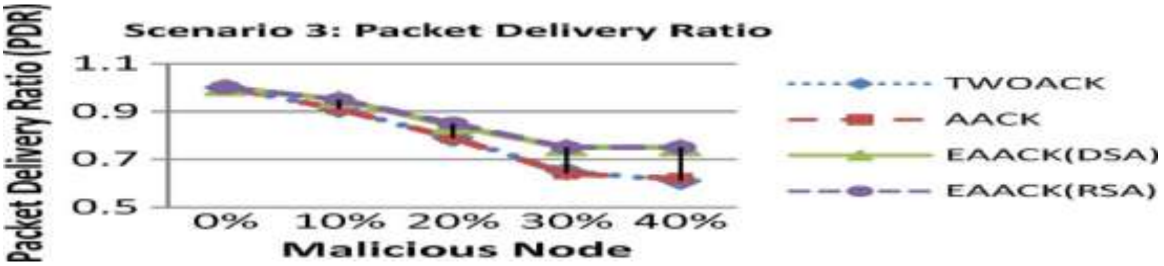


Fig. 13. Simulation results for scenario 3—PDR.

3) Simulation Results—Scenario 3: In scenario 3, we pro- vide the malicious nodes the ability to forge acknowledgment packets. This way, malicious nodes simply drop all the packets that they receive and send back forged positive acknowledgment packets to its previous node whenever necessary. This is a common method for attackers to degrade network performance while still maintaining its reputation. The PDR performance comparison in scenario 3 is shown in Fig. 13. We can observe that our proposed scheme RAACK outperforms TWOACK and AACK in all test scenarios. We believe that this is because RAACK is the only scheme which is capable of detecting forged acknowledgment packets.

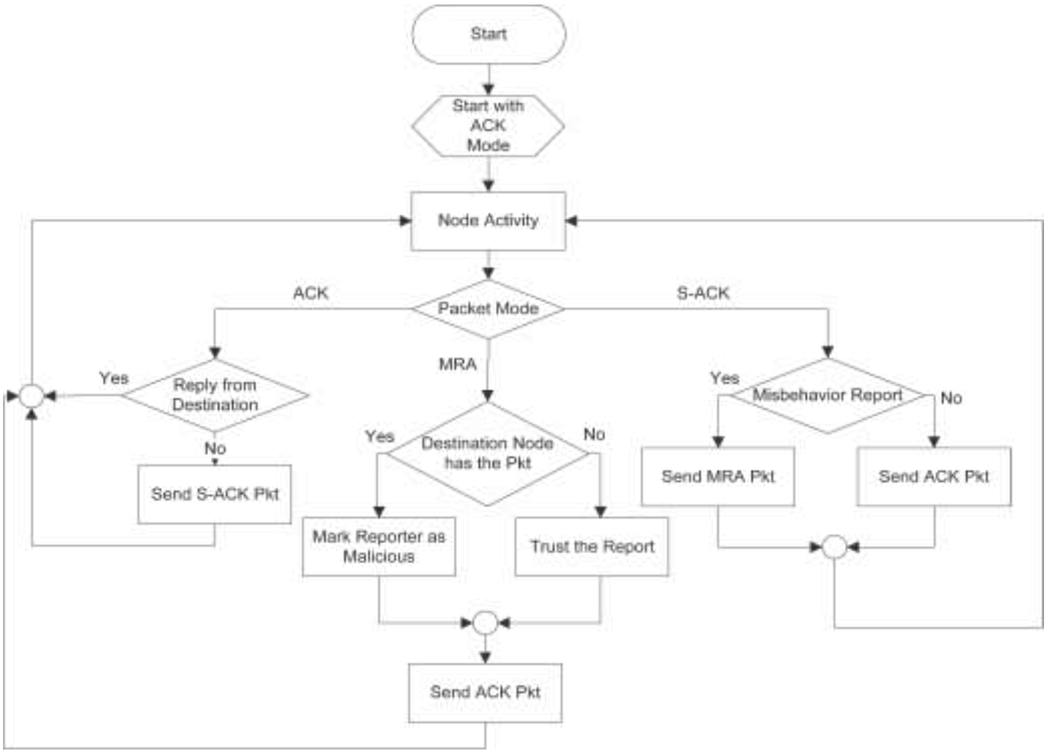


Fig. 14. S-ACK scheme: Node C is required to send back an acknowledgment packet to node A.

VI. CONCLUSION AND FUTURE WORK

Attack has always been a major threat to the security in MANETs. In this research paper, we have proposed a novel IDS named RAACK protocol specially designed for MANETs and compared it against other popular mechanisms in different scenarios through simulations. The results demonstrated positive performances against Watchdog, TWOACK, and AACK in the cases of receiver collision, limited transmission power, and false misbehavior report.

Furthermore, in an effort to prevent the attackers from initiating forged acknowledgment attacks, we extended our research to incorporate digital signature in our proposed scheme. Although it generates more rows in some cases, as demonstrated in our experiment, it can vastly improve the network’s PDR when the attackers are smart enough to forge acknowledgment packets. We think that this tradeoff is worthwhile when network security is the top priority. In order to seek the optimal DSAs in MANETs, we implemented ECDSA schemes. The key generated by the implementation is highly secured and it consumes lesser bandwidth because of small key size used by the elliptic curves. Significantly smaller parameters can be used in ECDSA than in other competitive systems such as RSA and DSA but with equivalent levels of security. Eventually, we arrived to the conclusion that the ECDSA scheme is more suitable to be implemented in MANETs. We are able to create a MANET network topology with minimum of 5 nodes to maximum of 8 nodes. We are able to find the shortest path and also identify all different paths from that source to destination..We are able to implement all the three modes like ACK, SACK, MRA scheme efficiently.

We are able to identify the malicious nodes which are in the network easily using RAACK To increase the merits of our research work, we plan to investigate the following issues in our future research:

1) Testing the performance of RAACK in software simulation instead of network environment.
2) We need to implement RAACK in a larger scale (LAN, WAN) and find out if RACCK works and is able to find out malicious nodes.
3) Implement on the various other network topologies other than what we have implemented till now.

REFERENCES

[1] K. Al Agha, M.-H. Bertin, T. Dang, A. uitton, P. Minet, T. Val, and J.-B. Viollet, “Which wireless technology for industrial wireless sensor networks? The development of OCARI technol,” IEEE Trans. Ind. Elec- tron., vol. 56, no. 10, pp. 4266–4278, Oct. 2009.

[2] R. Akbani, T. Korkmaz, and G. V. S. Raju, “Mobile Ad hoc Net-work Security,” in Lecture Notes in Electrical Engineering, vol. 127. New York: Springer-Verlag, 2012, pp. 659–666.

[3] R. H. Akbani, S. Patel, and D. C. Jinwala, “DoS attacks in mobile ad hoc networks: A survey,” in Proc. 2nd Int. Meeting ACCT , Rohtak, Haryana, India, 2012, pp. 535–541.

[4] T. Anantvalee and J. Wu, “A Survey on Intrusion Detection in Mobile Ad Hoc Networks,” in Wireless/Mobile Security. New York: Springer- Verlag, 2008.

[5] L. Buttyan and J. P. Hubaux, Security and Cooperation in Wireless Networks. Cambridge, U.K.: Cambridge Univ. Press, Aug. 2007.

[6] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, and L. Benini, “Model- ing and optimization of a solar energy harvester system for self-powered pp. 2759–2766, Jul. 2008.

[7] V. C. Gungor and G. P. Hancke, “Industrial wireless sensor networks: Challenges, design principles, and technical approach,” IEEE Trans. Ind. Electron., vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[8] Y. Hu, D. Johnson, and A. Perrig, “SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks,” in Proc. 4th IEEE Workshop Mobile Comput. Syst. Appl., 2002, pp. 3–13.

[9] Y. Hu, A. Perrig, and D. Johnson, “ARIADNE: A secure on-demand rout- ing protocol for ad hoc networks,” in Proc. 8th ACM Int. Conf. MobiCom, Atlanta, GA, 2002, pp. 12–23.

[10] G. Jayakumar and G. Gopinath, “Ad hoc mobile wireless networks routing protocol—A review,” J. Comput. Sci., vol. 3, no. 8, pp. 574–582,2007.

[11] D. Johnson and D. Maltz, “Dynamic Source Routing in ad hoc wireless networks,” in Mobile Computing. Norwell, MA: Kluwer, 1996, ch. 5, pp. 153–181.

[12] N. Kang, E. Shakshuki, and T. Sheltami, “Detecting misbehaving nodes in MANETs,” in Proc. 12th Int. Conf. iiWAS, Paris, France, Nov. 8–10, 2010, pp. 216–222.

[13] N. Kang, E. Shakshuki, and T. Sheltami, “Detecting forged acknowl- edgements in MANETs,” in Proc. IEEE 25th Int. Conf. AINA, Biopolis, Singapore, Mar. 22–25, 2011, pp. 488–494.

[14] K. Kuladinith, A. S. Timm-Giel, and C. Görg, “Mobile ad-hoc commu- nications in AEC industry,” J. Inf. Technol. Const., vol. 9, pp. 313–323, 2004.

[15] J.-S. Lee, "A Petri net design of command filters for semiautonomous mobile sensor networks," IEEE Trans. Ind. Electron., vol. 55, no. 4, pp. 1835–1841, Apr. 2008.

[16] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehaviour in MANETs," IEEE Trans. Mobile Comput., vol. 6, no. 5, pp. 536–550, May 2007.

[17] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbe- haviour in mobile ad hoc networks," in Proc. 6th Annu. Int. Conf. Mobile Comput. Netw., Boston, MA, 2000, pp. 255–265.

[18] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL: CRC, 1996, T-37.

[19] N. Nasser and Y. Chen, "Enhanced intrusion detection systems for dis- covering malicious nodes in mobile ad hoc network," in Proc. IEEE Int. Conf. Commun., Glasgow, Scotland, Jun. 24–28, 2007, pp. 1154–1159.

[20] J. Parker, J. Undercoffer, J. Pinkston, and A. Joshi, "On intrusion detection and response for mobile ad hoc networks," in Proc. IEEE Int. Conf. Perform., Comput., Commun., 2004, pp. 747–752.

[21] A. Patcha and A. Mishra, "Collaborative security architecture for black hole attack prevention in mobile ad hoc networks," in Proc. Radio Wireless Conf., 2003, pp. 75–78.

[22] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T. Karygiannis, "Secure routing and intrusion detection in ad hoc networks," in Proc. 3rd Int. Conf. Pervasive Comput. Commun., 2005, pp. 191–199.

[23] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 21, no. 2, pp. 120–126, Feb. 1983.

[24] J. G. Rocha, L. M. Goncalves, P. F. Rocha, M. P. Silva, and S. Lanceros Mendez, "Energy harvesting from piezoelectric materials fully integrated in footwear," IEEE Trans. Ind. Electron., vol. 57, no. 3, pp. 813–819, Mar. 2010.

[25] T. Sheltami, A. Al-Roubaiey, E. Shakshuki, and A. Mahmoud, "Video transmission nhancement in presence of misbehaving nodes in MANETs," Int. J. Multimedia Syst., vol. 15, no. 5, pp. 273–282, Oct. 2009.

[26] A. Singh, M. Maheshwari, and N. Kumar, "Security and trust management in MANET," in Communications in Computer and Information Science, vol. 147. New York: Springer-Verlag, 2011, pt. 3, pp. 384–387.

[27] B. Sun, "Intrusion detection in mobile ad hoc networks," Ph.D. disserta- tion, Texas A&M Univ., College Station, TX, 2004.

[28] K. Stanoevska-Slabeva and M. Heitmann, "Impact of mobile ad-hoc net- works on the mobile value system," in Proc. 2nd Conf. m-Bus., Vienna, Austria, Jun. 2003.

[29] A. Tabesh and L. G. Frechette, "A low-power stand-alone adaptive circuit for harvesting energy from a piezoelectric micropower generator," IEEE Trans. Ind. Electron., vol. 57, no. 3, pp. 840–849, Mar. 2010.

[30] M. Zapata and N. Asokan, "Securing ad hoc routing protocols," in Proc. ACM Workshop Wireless Secur., 2002, pp. 1–10.

[31] L. Zhou and Z. Haas, "Securing ad-hoc networks," IEEE Netw., vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.

[32] Botan, A Friendly C ++ Crypto Library. [Online].

[33] Nat. Inst. Std. Technol., Digital Signature Standard (DSS) Federal In- formation Processing Standards Publication, Gaithersburg, MD, 2009, Digital Signature Standard (DSS).

[34]TIK WSN Research Group, The Sensor Network Museum—Tmote Sky. [Online]. Available: http://www.snm.ethz.ch/Projects/TmoteSky

[35] Y. Kim, "Remote sensing and control of an irrigation system using a distributed wireless sensor network," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 7, pp.1379–1387,Jul. 2008