DOI: 10.21917/ijct.2010.0032

# HARDWARE IMPLEMENTATION OF SECURE AODV FOR WIRELESS SENSOR NETWORKS

**S. Sharmila[1], G. Umamaheshwari[2] and M. Ruckshana[3]**

[1]*Department of Electronics and Communication Engineering, Anna University of Technology, Coimbatore, India*
E-mail: hod@dit.psgtech.ac.in
[2, 3] *Department of Electronics and Communication Engineering, PSG College of Technology, India*

***Abstract***

***Wireless Sensor Networks are extremely vulnerable to any kind of routing attacks due to several factors such as wireless transmission and resource-constrained nodes. In this respect, securing the packets is of great importance when designing the infrastructure and protocols of sensor networks. This paper describes the hardware architecture of secure routing for wireless sensor networks. The routing path is selected using Ad-hoc on demand distance vector routing protocol (AODV). The data packets are converted into digest using hash functions. The functionality of the proposed method is modeled using Verilog HDL in MODELSIM simulator and the performance is compared with various target devices. The results show that the data packets are secured and defend against the routing attacks with minimum energy consumption.***

***Keywords:***
***Wireless Sensor Network, AODV, Routing, Security***

## 1. INTRODUCTION

A Wireless Sensor Network (WSN) is an aggregation of sensor nodes, distributed in an environment, to sense and collect information. A typical WSN consists of low-cost, low-power, and energy-constrained sensors responsible for monitoring a physical phenomenon and reporting to access points where the end-user can access the data. Wireless Sensor networks are widely used in tracking, security, area monitoring, industrial and health monitoring.

Current proposals for routing protocols [1]-[2] in sensor networks optimize the routing algorithms for limited capabilities of the nodes and the specific nature of the networks, but do not consider security issues. Although these protocols have not been designed with security as a goal, it is important to analyze their security properties. Sensor Nodes have insecure wireless communication, limited node capabilities and easily prone to internal threats. The adversaries can use high energy and long range communication to attack the network. Most network layer attacks such as Spoofed, Altered, or Replayed Routing Information, Selective Forwarding, Sinkhole Attacks, Sybil Attacks, Wormhole Attacks, Hello Flood Attacks and Acknowledgement Spoofing degrades the performance of the network. By spoofing, altering, or replaying routing information, adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, partition the network, increase end-to-end latency, etc.

Existing papers [3]-[6] address the defending mechanisms against routing attacks in algorithmic level by modifying the routing protocol and suggests the detection mechanisms. Researchers have proved high detection accuracy based on packet delivery ratio, Network throughput, communication overhead and routing overhead. The major limitations of the software approach are as follows:

1. Specific Detection Mechanism is required for each routing attack.
2. Communication Cost and Computational time is increased.
3. Additional memory for executing the detection algorithm is required.
4. Routing overhead is introduced and thus reduces the life time of the node while processing the control packets used for detection.

The motivation of this paper is to design the route processor to overcome the limitations of the exiting method. The wireless sensor nodes have limited life time, memory, and computational capabilities. The limitations of sensor nodes have to be considered while proposing a solution in hardware. Therefore, the main idea is to implement the route processor which in turn contains the routing and encryption process in hardware to prevent the routing attacks and to support reliable data delivery. The major feature of the proposed method provides unique solution to routing attacks. The security threats such as Integrity and Authenticity gives secure communication between the source node and destination node. The comparison of existing and proposed mechanism is shown in Fig.1. The rest of this paper is organized as follows.

In section 2, the hardware architecture of secure routing is explored. In section 3, Simulation results are presented. Finally, in the conclusion the scope of future work is presented.

## 2. SECURE AODV ROUTING PROTOCOL

The AODV is a reactive protocol; routes are created only when a node wants to communicate with another node. The primary objectives of AODV are to discover the path, to identify the destination and to transfer the encrypted packets between the source and destination. There are four types of control packets RREQ, REER, HELLO and RREP. The first three are sent by broad cast, while RREP is sent by unicast. It discovers the route based on local routing table. The algorithm enables dynamic, self starting, and multi-hop routing between the participating mobile sensor nodes.

The existing algorithm is implemented in software and requires more power consumption. It consumes longer processing time for execution of algorithm. The existing algorithm is insecure and it is prone to network routing attacks.

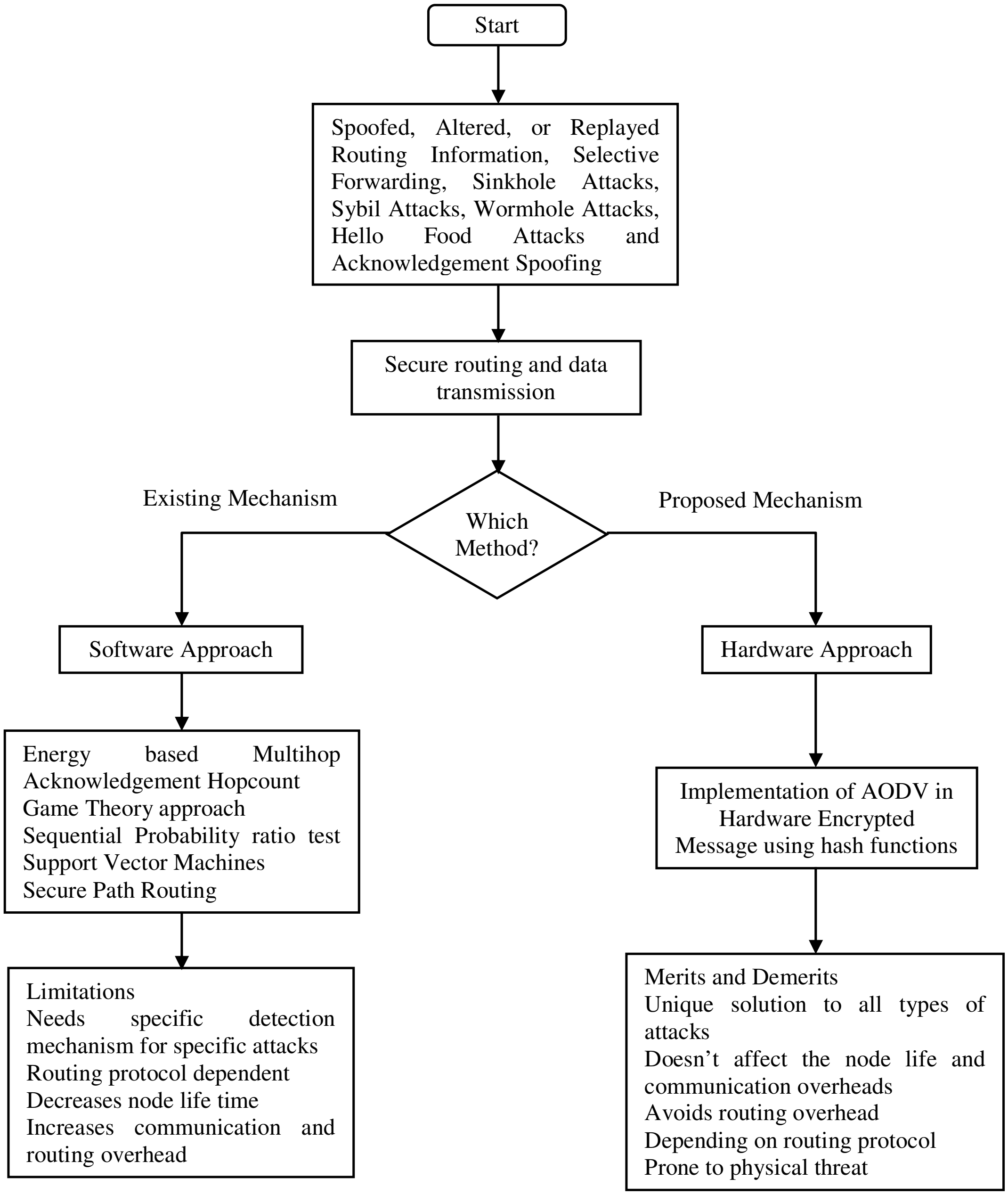
Start
Spoofed, Altered, or Replayed Routing Information, Selective Forwarding, Sinkhole Attacks, Sybil Attacks, Wormhole Attacks, Hello Food Attacks and Acknowledgement Spoofing
Secure routing and data transmission
Existing Mechanism
Which Method?
Proposed Mechanism
Software Approach
Hardware Approach
Energy based Multihop Acknowledgement Hopcount
Game Theory approach
Sequential Probability ratio test
Support Vector Machines
Secure Path Routing
Implementation of AODV in Hardware Encrypted Message using hash functions
Limitations
Needs specific detection mechanism for specific attacks
Routing protocol dependent
Decreases node life time
Increases communication and routing overhead
Merits and Demerits
Unique solution to all types of attacks
Doesn't affect the node life and communication overheads
Avoids routing overhead
Depending on routing protocol
Prone to physical threat

Fig.1. Comparison of Detection Mechanism using Hardware and Software approaches

Fig.2. represents the basic design approach for secured AODV routing protocol. Initially, the node is in ideal state. When an originator needs to communicate with another node, namely the destination, the source of the originator broadcasts the RREQ message. Once the destination is reached, it generates RREP message which is unicast to the source. It is known that a node can be a source, an intermediate node and a destination. When a node in an active route gets lost, a route error message (RERR) is generated to notify the other nodes on both sides of the link of the loss of the link.

Route discovery is initiated by broadcasting a RREQ message. The route is established when a RREP message is received. A source node receives multiply RREP messages with different routes. It then updates its routing entries if and only if the RREP has a greater sequence number, i.e. fresh information. Reverse path setup is established while transmitting RREQ messages through the network each node notes the reverse path to the source. When the destination is found the RREP message will travel along this path, so no more broadcasts will be needed.

When a broadcast RREQ packet arrives at a node having a route to the destination, the reverse path will be used for sending a RREP message. While transmitting this RREP message the forward path is set up. As soon as the forward path is built the data transmission can be started. Data packets waiting to be transmitted are buffered locally and transmitted in a FIFO-Queue when a route is set up. After a RREP was forwarded by a node, it can receive another RREP.

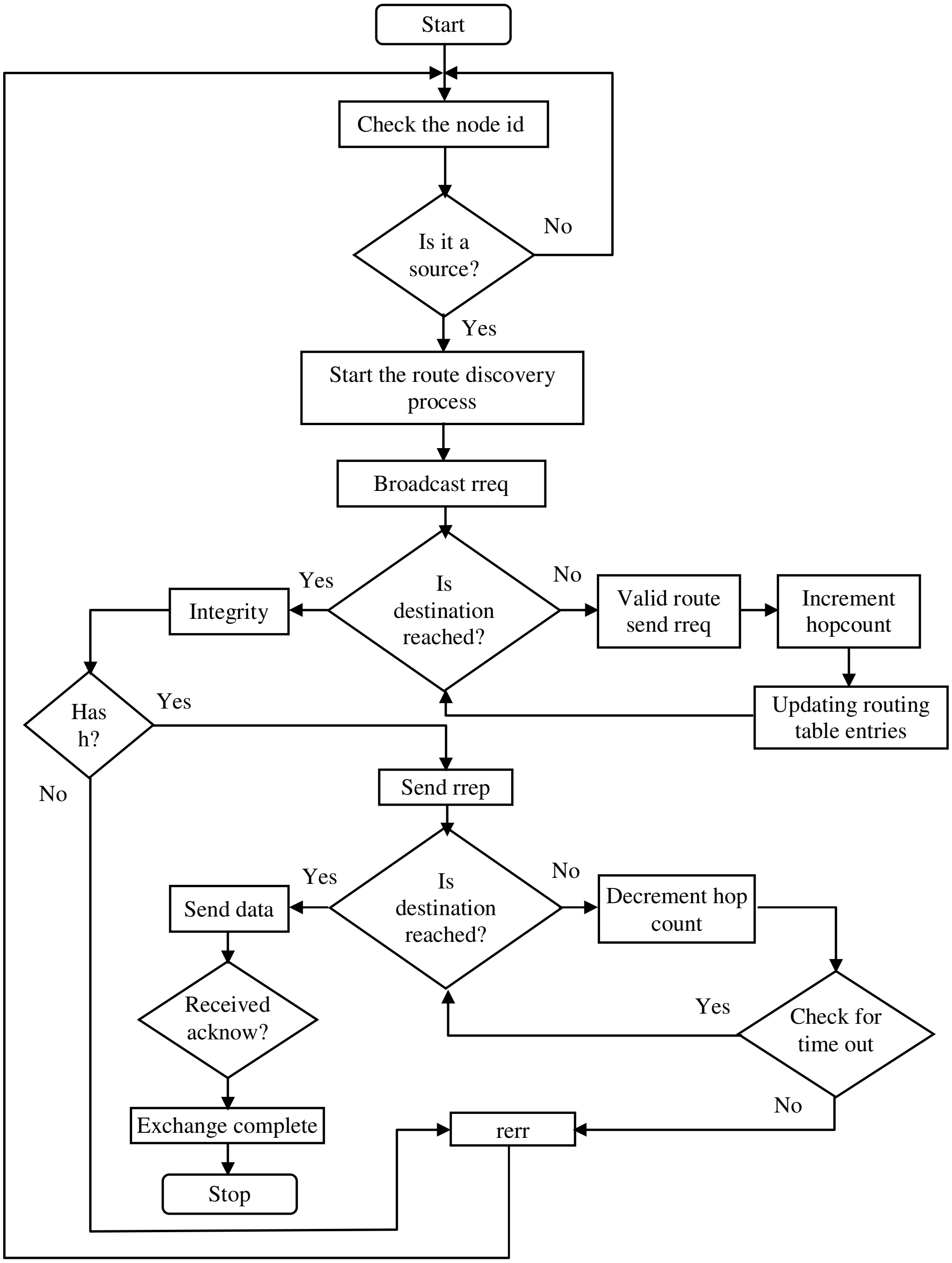

Fig.2. Secure AODV Routing Protocol

This new RREP will be either discarded or forwarded, depending on its destination sequence number:

- If the new RREP has a greater destination sequence number, then the route should be updated and RREP is forwarded
- If the destination sequence numbers in old and new RREPs are the same, but the new RREP has a smaller hop count , this new RREP should be preferred and forwarded
- Otherwise all later arriving RREPs will be discarded.

In the proposed method, security has been added to prevent the packet transmission from malicious node attack. This is done by padding the 22-bit packet with additional bits so that the original packet being transferred to the neighbor node identifies the malicious node as 20-bits. This encrypted data is again decoded and combined with the original packet being transferred. If both are same, then the node is considered to be in the sensor network, else it is noted to be a malicious node.

After finding it as error free, the path between the original node and the destination is established. The source in turn will send the data packets to the destination.

There are various control packet formats. A 22-bit packet is taken in this case for a miniature model since the real time data packets are of 192-bits in length. The packet formats considered for the design is shown in Fig.3, which is used throughout the design.

## 2.1 ARCHITECTURE OF SECURED AODV ROUTING PROTOCOL

The basic functional block diagram of the secured AODV routing protocol for mobile wireless sensor nodes is given in Fig.4. A wireless sensor node can be either stationary or mobile. The data is routed from the source to destination through a routing protocol. Since node size is small and has limited power, a unique architecture is required for secure routing. The architecture contains processor, input buffer, output buffer, cache memory, control unit, integrity and FIFO memory. The purpose of I/O buffer is to store the incoming and outgoing message packets. The purpose of cache memory is used to store the addresses of the adjacent nodes to forward the packet. This is implemented in the form of matrix representation. The FIFO is used to store the address of the nodes, which lies in the forward path. The hardware contains comparator, decoder, multiplexer, latch, and flip-flops for route discovery and data integrity. The architecture of the secured AODV routing protocol is given below in Fig.4.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RCAK | TYPE (4) | RESERVED (18) | | | | | |
| DATA | TYPE (4) | RESERVED (18) | | | | | |
| RREQ | TYPE (4) | HOPCNT (3) | RREQ (3) | DID (3) | DSN (3) | OIP (3) | QSN (3) |
| RREP | TYPE (4) | HOPCNT (3) | DID (3) | DSN (3) | OIP (3) | LIFETIME (6) | |
| RERR | TYPE (4) | DESTNCNT (3) | UNREACH DESTN (3) | UNREACH DSN (3) | RES (9) | | |

Fig.3. Message packets for AODV

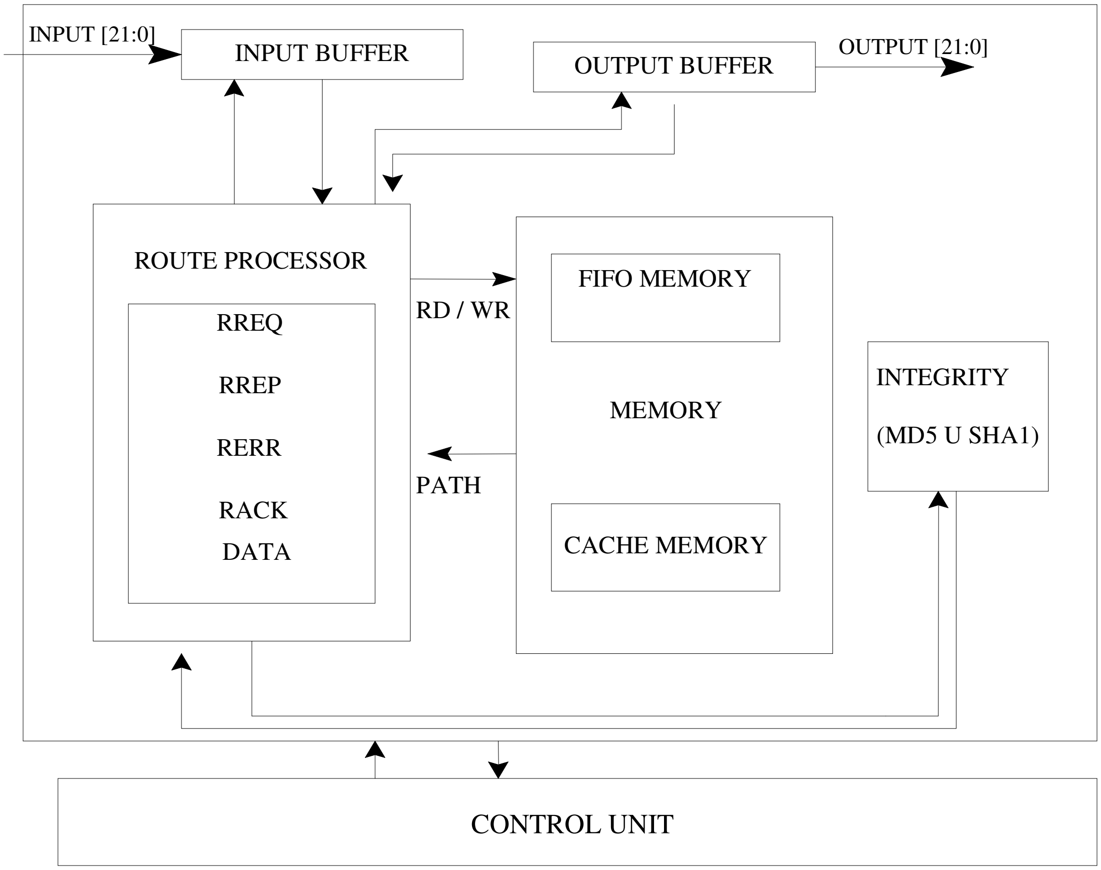

Fig.4. Hardware Architecture of Secured AODV Routing Algorithm

The Route Request, Route Reply, Route Acknowledgement, data exchange, and probing message signals are controlled by the control and data path unit. The input packet contains type [4-bits] i.e. to identify whether it is a RREQ, RREP, RACK or RERR. The hop count [3-bits], is incremented if the node, is an intermediate node. The rreq id [3-bits] contains the address of the source node. The destination id [3-bits] represents the destination nodes address. The destination sequence number [3-bits] is used to check whether the node is an intermediate or not. The organization ip[3-bits] and organization sequence[3-bits] are also present. From the message packet received only the destination address alone is removed and the path to reach the destination from the source is tracked. Based on the data received the message packet is altered and the process is repeated. The Table.1 describes the bit change for the control signals being used for the method. Based on these control signals the packet functions are identified.

Table.1. Control Packets

| Control Packets | Data-bits |
|---|---|
| RREQ | 0000 |
| RREP | 0001 |
| RERR | 0100 |
| RACK | 0010 |
| DATA | 1111 |

The functioning of the memory unit is as shown in Fig.5. Initially the message is stored in the FIFO queue and it is interpreted to identify the current state of the node using automatic control table. The identified current state is validated to route the packets for next hop. The index and the address of the next hop are stored in the automata state table. The routing table is updated in table to decide the route for the next hop.

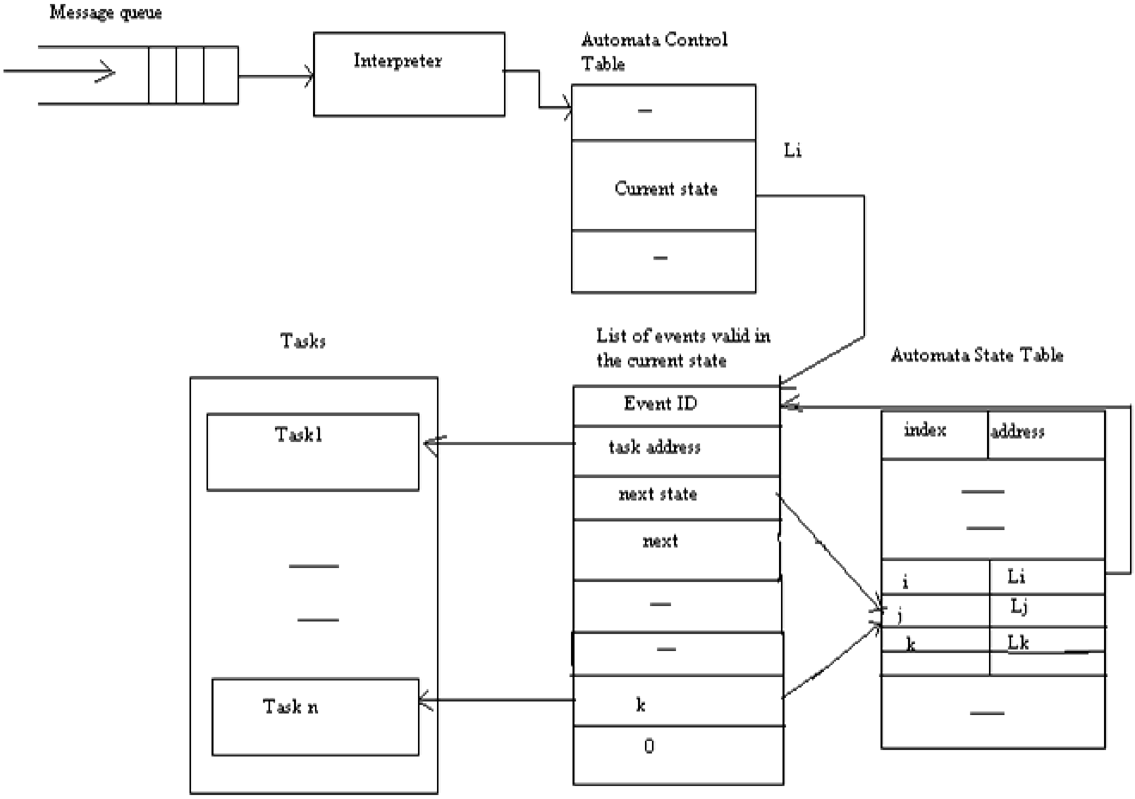

Fig.5. Functional Representation of Memory

## 2.2 INTERNAL ARCHITECTURE OF ROUTE PROCESSOR

The route processor performs the operation of various control packets present in AODV [7]-[8]. The control packets are RACK, DATA, RREQ and RERR. The functioning of these control packets is described in this section.

### *2.2.1 Data exchange*

A 22 bit packet is taken in this case for a miniature model since the real time data packets are of 192 bits in length. The incoming data packet consisting of 21 bits is stored in the input buffer. The first 4 bits are extracted and e applied into the de-multiplexer unit from which it is decided whether it is route request, route reply, route acknowledgement, data exchange or route error request. Based on these control signals the rest of the blocks are enabled. If the incoming packet is a route request, then by use of comparator block, the destination address of the packet is compared with nodes destination address, if both are equal then the RREP message is send to the source node. After receiving the route acknowledgement from the source node, the source nodes sequence number of the source node is compared with the destination nodes sequence number. If they are equal then the packet is applied to the integrity module. The above architecture is shown in Fig.6.

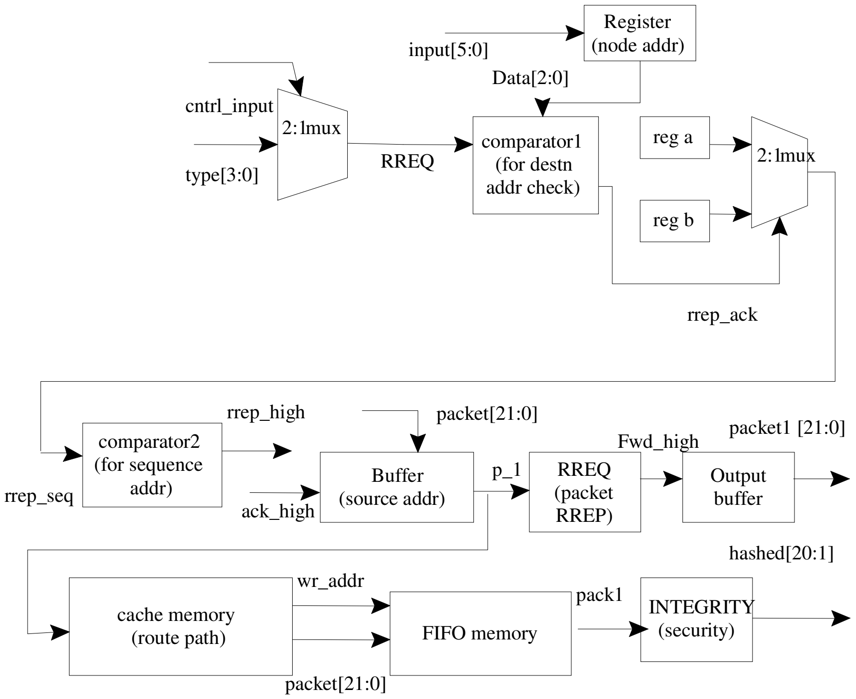

Fig.6. Architecture of Data Exchange

The integrity module being used is the MD5USHA1 cryptographic function. The module consists of buffers, XOR gates, registers and adder units. The input to the integrity module is 512 bits. The input packet from the source node consisting of 22 bits is again padded with the remaining bits and is then forwarded to the source node along with a hit control signal.

The source node on receiving the control signal generates hashed value based on the packet being sent to the destination node. Then, both the secured data from the destination node and the source node are compared using a 20-bit comparator. If both are equal, then an hash-acknowledgement signal is given to the destination node along with the data packet to be transferred. The destination node on receiving the data packet along with the control signal stores the data packet in FIFO memory and sends the acknowledgement to the source node, as data exchange is complete.

### *2.2.2. Route Request (RREQ)*

If the RREQ is received, the destination address is compared with the node address. If equal, then the above mentioned process is repeated, if not equal then destination address is sent to the cache memory which contains the routing information of the adjacent nodes. If the route to the destination node is found, then the packet is transferred as RREQ packet to its adjacent nodes. Original node address, is stored in the buffer for route reverse path. This is shown in Fig.7.

### *2.2.3 Route Reply (RERR)*

If the packet being received is a RREQ packet, then by use of comparator block the destination address of the packet is compared with node destination address, and if both are equal then the RREP message is sent to the source node. After route acknowledgement from the source node, the source nodes sequence number is compared with the destination nodes sequence number and if equal the packet is then applied to the integrity module.

The integrity block generates a hashed value for the above packet. The control signal is then sent to the source node. The source node generates the hashed value if both values are equal and the packet is forwarded. If not equal then the source nodes decides it as a malicious node and a err-acknowledgement is transferred to the malicious node. The source node also removes the neighbor node address from the route path. Thus the malicious node is being identified. This is shown in Fig.8.

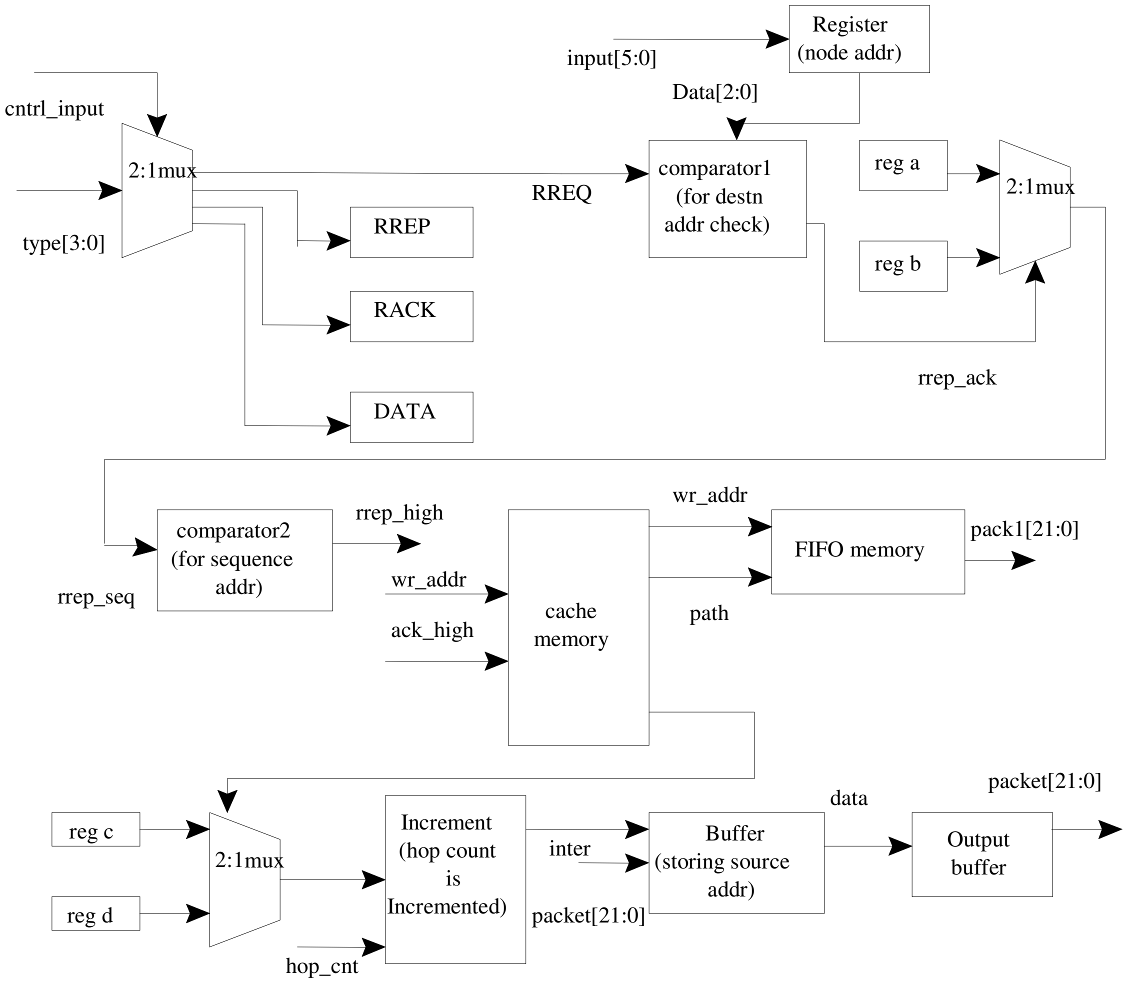

Fig.7. Architecture of RREQ

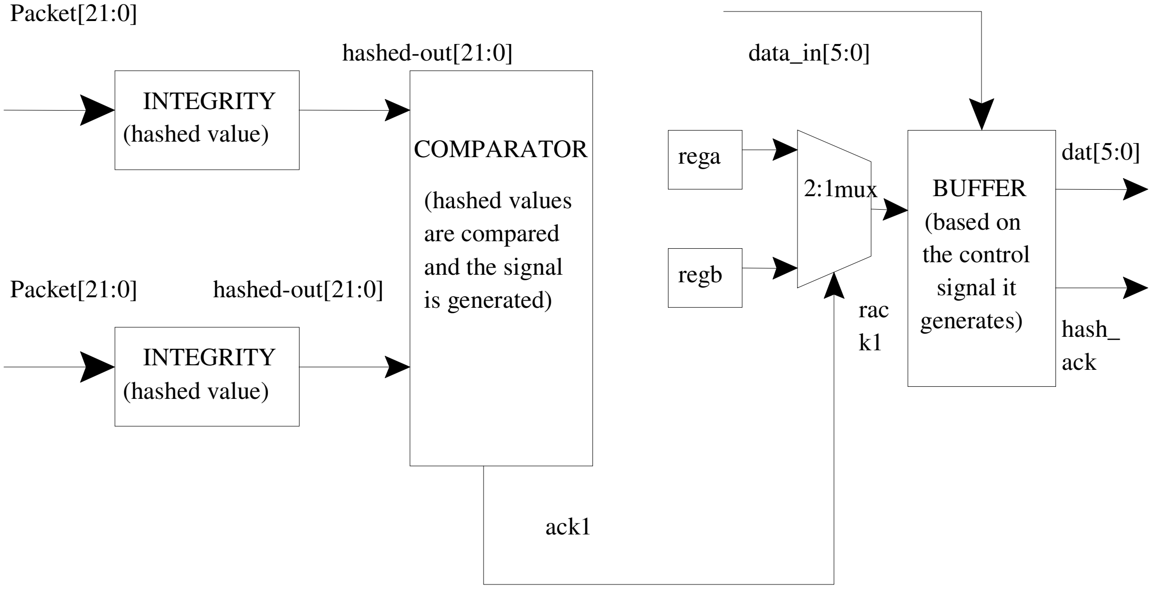

Fig.8. Architecture of RERR

### *2.2.4 Message Encryption*

A single block of message of size 512 bits is passed through secure hash algorithm first and then through message digest algorithm [9]. Both the algorithms (extended message digest algorithm and secure hash algorithm) have four rounds and each round has twenty steps.

The description of single step of both the algorithms is given in Fig.9. A single step of the combined architecture involves processing through secure hash algorithm first and then through extended message digest algorithm.

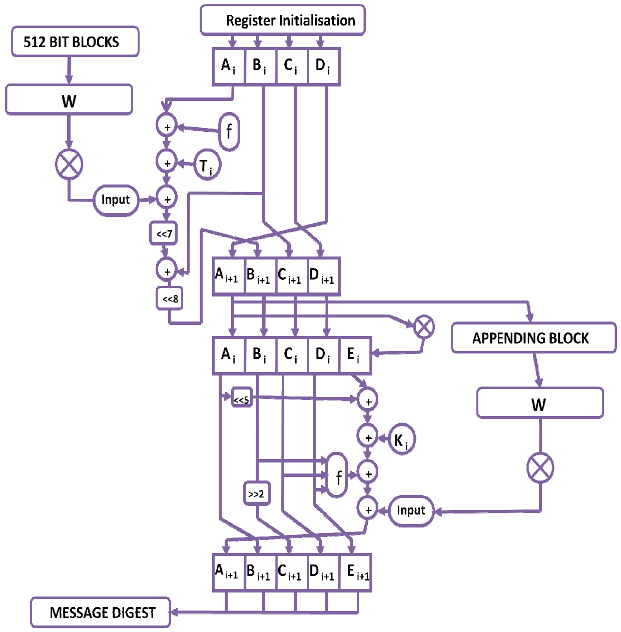

Fig.9. Combined Architecture of MD5 U SHA1

The first process is passing the message through the single step of the SHA1 algorithm and the resultant values in the chaining variables are shuffled as per the algorithm. The chaining variables are passed through the extended MD5 algorithm where the extra 32 bit word ‘e’ is retained in the same variable as such.

Then, the second step is continued. The single steps of both the digest algorithms are same as that of the original algorithms except for the extended message digest algorithm. The concatenated chaining variables in the end give the hash calculated for the first block of message. This digest is then used as the initial value for the next block of message. In the end the digest for the complete message is obtained. It is observed that the collision resistant is improved when compared with the existing method.

### *2.2.5 Control Unit*

The state modeling of secured AODV protocol can be seen below which reveals the basic process undergone by the AODV router (ie) processing type of messages, route request, route reply etc. The initial state was taken to be the idle state where the nodes are in reset condition without any transactions but is logged to the network.

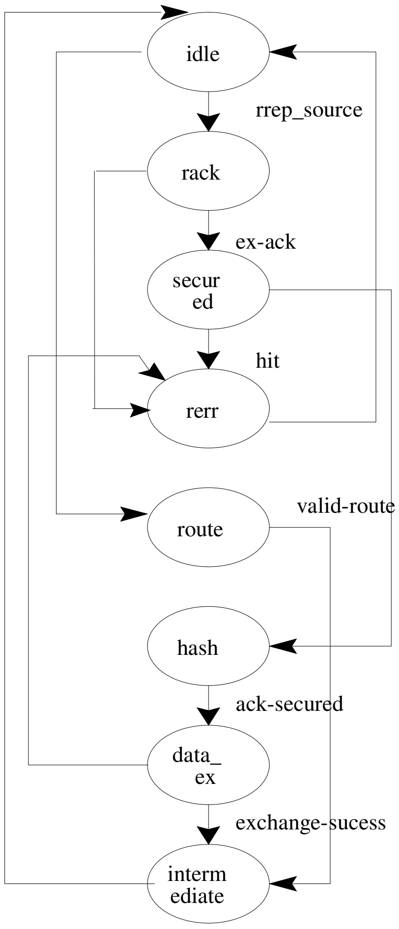

Fig.10. Finite State model of AODV Routing Protocol

A node can be a source, destination or an intermediate node, and this condition is considered in the beginning. This protocol is implemented using 8 states through which the control signals are transferred to the block and thus both the data path unit and the control unit functions. The Finite State Machine is described in the Fig.10.

## 2.3. FUNCTIONALITY CHECK

In the above architecture, the following changes occur in the output for the given input conditions. This is shown in the Table.2.

Table.2. Functionality Check

| Input | Output |
|---|---|
| RREQ (0000) | RREP (0001) or RREQ (0000) or RERR (0100) |
| RREP (0001) | RACK (0010) or RERR (0100) |
| RACK (0010) | DATA (0010) or RERR (0100) |
| DATA (1111) | RACK (0010) or RERR (0100) |

## 3. SIMULATION AND RESULTS

The above architecture is implemented using VHDL program and simulated. Fig.11. shows the RTL Schematic of the control signals being generated for the entire architecture. The processor generates the control signals for finding the data path. The RREQ, RREP, RACK and RERR must be HIGH.

The input applied is RREQ since the received node is an intermediate node and the output being obtained is RREQ, which is flooded to the next node and is shown in Fig.12. The RREP is shown in Fig.13. The input applied is RACK since the packet being received is not the required destination node the output packet being obtained is RERR and is shown in Fig.14.

Fig.15. shows the snapshot of Architecture after routing. The proposed architecture is taken up to the ASIC flow level design in which the floor planning and routing of the design is obtained. The area required for the proposed method for implementation in chip level is obtained as Area (um$^2$) = 18319.The power required for the computation is obtained using X Power Analyzer in ISE. The power dissipated for the target device (virtex4), is = 216mw. The power needed for the proposed method is Total Dynamic Power(mw) = 420.5457 & Cell Leakage Power(µw) = 53.3523.

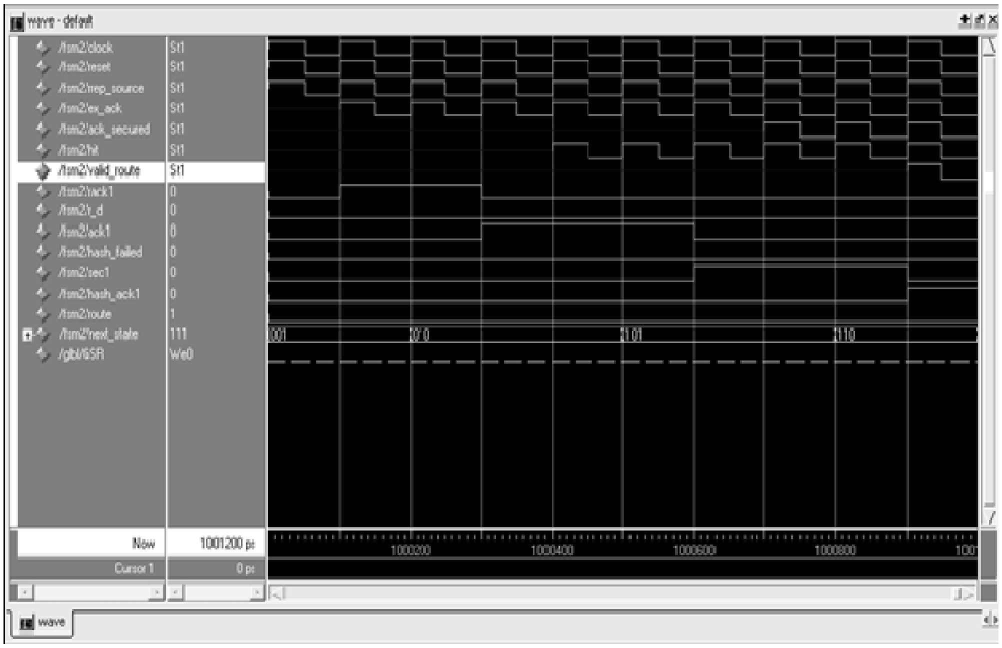

Fig.11. Control Signals

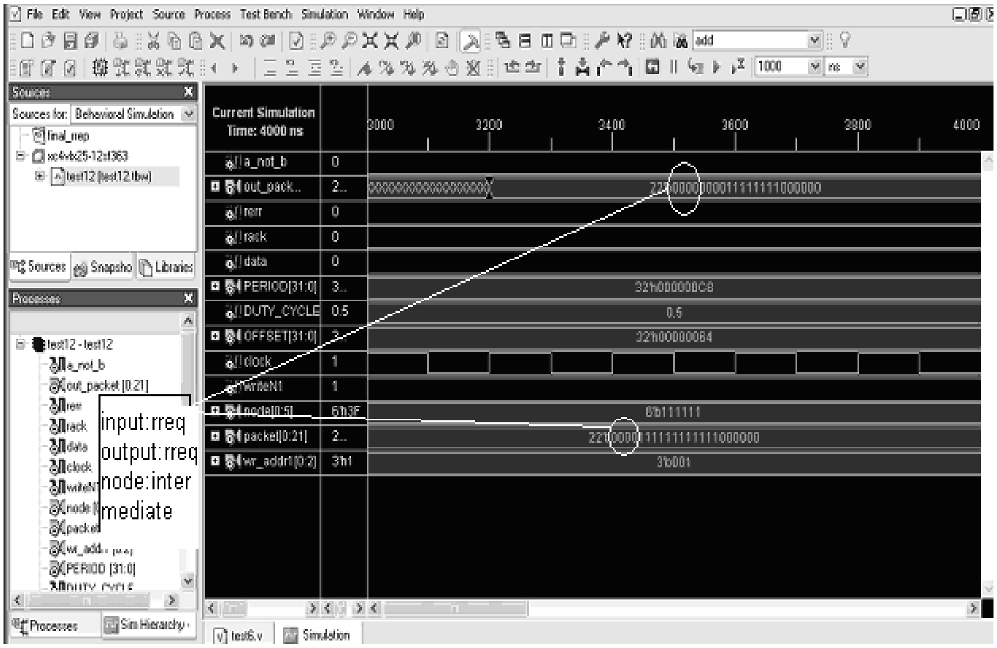

Fig.12. Route Request Message

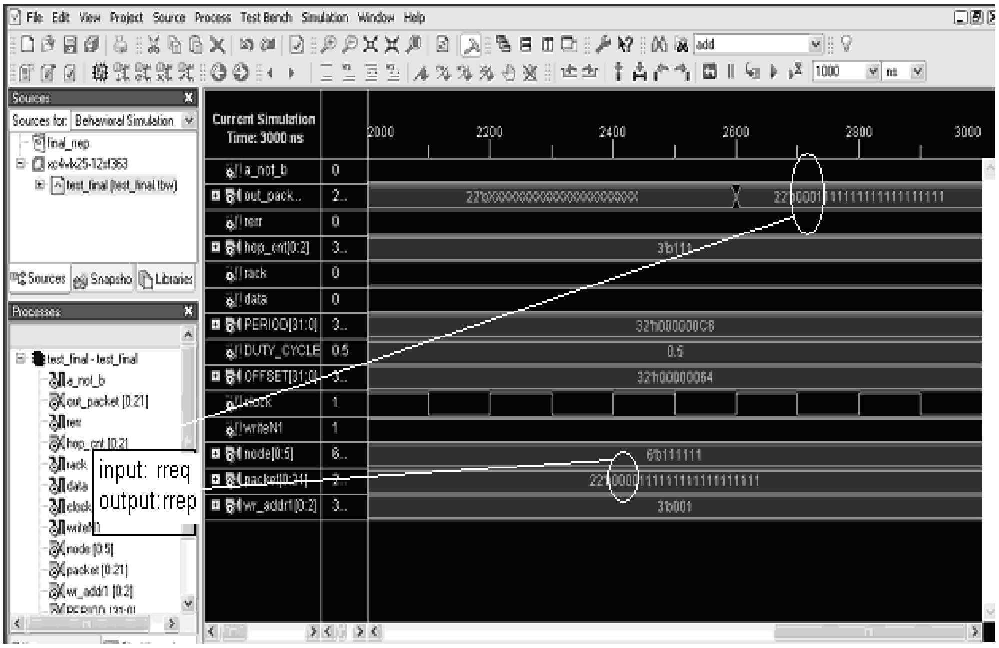

Fig.13. Route Reply Message

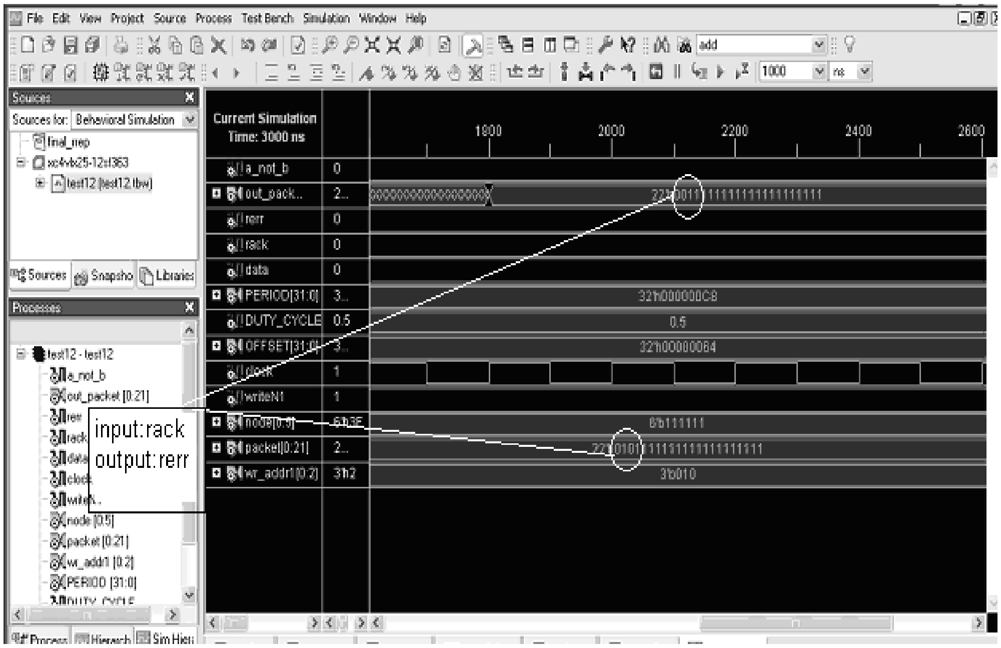

Fig.14. Route Error Message

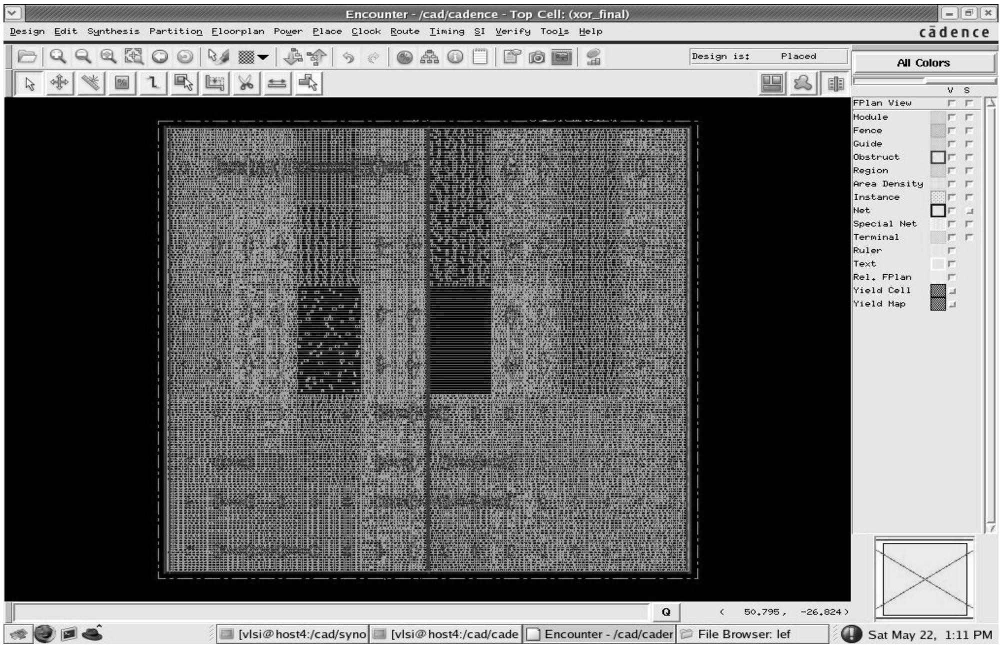

Fig.15. Snapshot of Architecture after routing

Table.3. Comparison between the Existing and Proposed Scheme

| Parameters | AODV [12] | Secure AODV |
|---|---|---|
| Maximum Frequency | 75.069MHz. | 80MHz |
| Selected Device | 4Vlx25sf363-12 | 4vlx25sf363-12 |
| Number of Slices | 55% | 48% |
| Number of Flip Flops | 17% | 10% |
| Number of 4 input LUTs: | 50% | 44% |
| Number of GCLKs: | 6% | 4% |
| Connection Establishment time | Minimum | Minimum |
| Topology | Dynamic | Dynamic |
| Speed and Efficiency | Improved | Improved |
| Security | Provided | Not Provided |
| Routing attacks | Prone to attacks | Against the selective forward , sink hole and warm hole attacks |
| Time delay | Increased | Decreased |
| Dynamic Power(mw) | Not Analyzed | 420.5457 |
| Cell Leakage Power (µW) | Not Analyzed | 53.3523 |
| Data Integrity | Not included | Hash Functions |

The proposed method is compared with the existing method and is shown in Table.3. It is observed that the proposed method is an efficient method for routing the packets in mobile wireless sensor networks. Since routing is done in hardware, there may not be any possibility for intruder to know about the hop counts and routing information. The integrity is achieved through hash functions. The Mica mote is a small (several cubic inch) sensor/actuator unit with a CPU, power source, radio, and several optional sensing elements. The processor is a 4 MHz 8-bit Atmel ATMEGA103 CPU with 128 KB of instruction memory, 4 KB of RAM for data, and 512 KB of flash memory. The CPU consumes 5.5 mA (at 3 V) when active, and two orders of magnitude less power when sleeping. Comparing the power consumed for the simulation, the algorithm is suitable for implementation in the MICA motes.

## 4. CONCLUSIONS

The proposed hardware architecture of secured AODV routing protocol in Wireless sensor node provides a novel implementation technique that is invented using Virtex IV device from Xilinx family. This architecture reacts quickly to dynamic topology changes even under high traffic. The speed and efficiency of the node is increased. The proposed method is against network routing attacks. In this method message integrity is achieved using the combined architecture of MD5USHA5. From the analysis, the message digests provide a secure way to check for alteration in the message packets that are transferred. The delay and power of the proposed method is compared with that of existing AODV done in hardware level, where security is not considered. Also, the algorithm is found suitable for implementation in the MICA motes. In this proposed method, routing protocol cannot be changed as per the topology condition. In future, the proposed method is optimized further to reduce time delay and it can be implemented in FPGA, that can be used as add on card for real time applications where wireless sensor nodes are used.

## REFERENCES

[1] S. Keshav and R. Sharma, 1998, "Issues and Trends in Router Design", The IEEE Communication Magazine, Vol.36.

[2] Chris Karlof and David wagner, 2003, "Secure routing in wireless sensor networks: attacks and countermeasures", The IEEE International Workshop on Sensor Network Protocols and Applications, pp. 113-127.

[3] Amitabh Mishra, Ketan Nadkarni, and Animesh Patcha, 2004, "Intrusion Detection in Wireless Ad Hoc Networks", IEEE Wireless Communications, Vol.11, No.1, pp. 48 – 60.

[4] Hao Yang and Haiyun Luo, 2004, "Security in Mobile Ad Hoc Networks: Challenges and Solutions", IEEE Wireless Communications, Vol.11, No.1, pp. 38 – 47.

[5] M.Ramakrishnan and S Shanmugavel, 2008, "New Approaches to Routing Techniques of MANET Node for Optimal Network Performance", The International Journal of Computer science and Network Security, Vol.8, No.11, pp. 369-376.

[6] Farid Nait-Abdesselam and Brahim Bensaou, 2008, "Detecting and avoiding wormhole attacks in wireless Ad hoc networks", IEEE Communication Magazine, vol.46, No. 4, pp.127 – 133.

[7] Junajo Noguera and Rosa M.Badia, 2002, "HW/SW co design Techniques for Dynamically reconfigurable architectures", IEEE Transactions on Very large Scale Integration Systems, Vol.10, No.4, pp.399-414.

[8] Ali El Kateeb, Aiyappa Ramesh and L.Azzawi, 2008, "Wireless Sensor Node Processor Architecture and Design", in Proc.CCECE2008.

[9] H. Mirvaziri, Kasmiran Jumari, Mahamod Ismail and Z. Mohd Hanapi, 2007, "A new Hash Function Based on Combination of Existing Digest Algorithms", *in Proc***.** IEEE conference (EEI).