Development of an Improved Intrusion Detection based Secured Robust Header Compression Technique

Malachi C. Egbugha
Electrical and Computer Engineering Department, Ahmadu Bello University, Nigeria

I. J. Umoh
Electrical and Computer Engineering Department, Ahmadu Bello University, Nigeria

A. M. S. Tekanyi
Electrical and Computer Engineering Department, Ahmadu Bello University, Nigeria

ABSTRACT
This research presents the development of an Improved Intrusion Detection Secured RObust Header Compression (IDSROHC) technique for handling brute force attack. The Secured RObust Header Compression (Secured ROHC) was developed to secure internet protocol version six (IPv6) packets against false initial refresh attack by encrypting the cyclic redundancy check field. However, the CRC is only 3-8 bits long, which implies that a malicious node could still attempt a brute force approach, where it sends fake packets with all possible CRC combinations.An IDSROHC was developed using a modified selective watchdog intrusion detection algorithm. A MATLAB graphical user interface was design to aid presentation. IDSROHC was validated with Secured ROHC using throughput and packet delivery success. The results of this work show that IDSROHC produced 4.97% improvement in throughput and 29% improvement in packet delivery success over Secured ROHC.

Keywords
IDSROHC, Secured ROH,Brute force-Attack, Throughput, Packet delivery Success

1. INTRODUCTION
The desire for industries to move towards an Internet Protocol Version six (IPv6) network architecture has pushed research in the direction of maximizing bandwidth. This is due to increased header size of IPv6 header as compared to the payload .Therefore, reducing the internet protocol header overload sent over the air becomes inevitable [1]**.** One method of providing increased bandwidth efficiency is the use of IP header compression techniques. Header compression provides more efficient use of bandwidth in a packet switched network by taking advantage of header field redundancies in packets belonging to the same flow [2]. To further increase the bandwidth efficiency, the sliding window can be made adaptive with respect to packets loss [3]. IP Header compression involves a compressor and a decompressor operating according to a well-defined protocol. The compressor compresses the headers with respect to a reference state that it shares in common with the decompressor, while the decompressor uncompresses them to their original state on reception at the destination [4]. Header compression technique falls into two major categories: stateful header compression and stateless header compression.The stateful header compression technique builds hop-by-hop compression per flow and requires state management.These include Van Jacobson Header Compression (VJHC), RObust Header Compression (ROHC) and Internet Protocol Header Compression (IPHC) .Stateless header compressions such as Mobile adhoc network Internet Protocol Header Compression (MIPHC) does not require state management [5]. The ROHC is designed to operate efficiently and robustly over various link technologies with different characteristics [6]. While this exchange leads to efficient bandwidth utilization, there are several potential attack such as False Initialization/Refresh (False IR), False ACKnowledgment (False ACK) and False Negative ACKnowledgment (False NACK) attack that can lead to denial of service (inability to decompress) [1]. In other to solve this problem, research has focused on cryptographic method such as Cyclic Redundancy Check (CRC) encryption [1] [7]. However, due to limited number of bits, a malicious node could still attempt a brute force approach where it sends fake packets with all possible cyclic redundancy check combinations therefore resulting in decryption of the cyclic redundancy check.

In any network security plan, if intrusion prevention (encryption, authorization, and authentication) is defeated by attackers, then a second line of defence, intrusion detection comes into prominence [8]. Intrusion detection provides deterrence for an intruder and serves as an alarm mechanism for a computer system or a network to manage a security plan successfully. An Intrusion-Detection System (IDS) is defined as a software or hardware monitoring tool that detects internal or external cyber-attacks. An IDS can observe and investigate system and user activities, recognize patterns of known attacks and identify abnormal network activity. An IDS developed using a modified selective watchdog technique was therefore employed in this research to detect and mitigate brute force attack in a Secured robust header compression network [9].

2. LITERATURE REVIEW
2.1 RObust Header Compression (ROHC) Scheme
IP header compression is the process of reducing protocol header overhead in order to improve bandwidth efficiency while maintaining the end-to-end transparency [10, 11]. IP header compression concept relies on the characteristic that many header fields in consecutive packets belonging to the same packet flow remain a constant or change in predictable manner [11, 12].

VJHC and IPHC protocols were the first IP header compression scheme created. The IPHC scheme was created to extend the work done in VJHC. However, it was not robust enough to support links with high bit error rates, high losses, and long round trip times. High Bit Error Rate (BER) and long Round Trip Time (RTT) are common characteristics of wireless links. Therefore an efficient and robust compression scheme was needed. The ROHC scheme was developed to fulfil these criteria [13, 14]. It is a standard approach suitable for links with significant error rates and long round-trip time [14]. The ROHC scheme uses window based least significant bits encoding for the compression of dynamic fields in the protocol headers. Due to its feedback mechanism, periodic context refreshes and

Window-based Least Signicant Bits (W-LSB) encoding scheme, ROHC is robust on wireless links with high BER and long RTT [11]. The W-LSB is defined by an interpretation interval. Interpretation interval is the maximum number of packets that is lost in a row without losing context synchronization [15]. It is expressed mathematically as follows [16]:

$$W = [-o_f, 2^b - 1 - o_f] \quad (1)$$

where:

W is the interpretation interval

o_f is the offset

b is the least significant bit

ROHC header compression framework is a process of interaction between two state machines; a compressor state machine and a decompressor state machine described by their context [17]. Both the compressor and the decompressor maintain three interrelated states with slight differences.

2.2 Random waypoint mobility model

In mobility management, the random waypoint model is a random model for the movement of mobile users, and how their location, velocity and acceleration change over time. Mobility models are used for simulation of users movement pattern. The random waypoint mobility model was first proposed by Johnson and Maltz. In random waypoint a node randomly chooses a new destination (x; y) within a given playfield area and a speed that is uniformly distributed between a minimum and maximum speed [minspeed; maxspeed]. Upon arrival at the destination, it pauses for a specified time (Tpause) period after which it again selects a random destination and speed and continues likewise. In the Random Waypoint model, maxspeed and Tpause are the two key parameters that determine the mobility behavior of nodes. If Tpause=0, this leads to continuous mobility. In addition, Long Tpause and small maxspeed leads to stable topology. In ROHC, the distance between two nodes (Compressor and Decompressor) using random way point model is thus calculated as follows [18]:

$$R = \sqrt{(X_d - X_s)^2 + (Y_d - Y_s)^2} \quad (2)$$

where:

R is the distance between the compressor (source node) and decompressor (destination node)
X_s and Y_s are the Cartesian coordinate of compressor (source node)
X_d and Y_d are the cartesian coordinate of decompressor [19].

2.3 RObust Header Compression (ROHC) Model

One of interesting properties of W-LSB is that decompression will not fail unless the number of consecutive packet losses exceeds the interpretation interval. Hence, interpretation interval plays an important role in the choice of W-LSB and has a direct impact on the performance of ROHC. A small interpretation interval generates fewer bits, has a faster decompression speed and shorter compressed header than a large interpretation interval. However, small interpretational interval leads to higher compression efficiency and low robustness compared to large interpretational interval. Therefore, for small number of packet losses, a small interpretation interval is sufficient. When large packet loss is expected (that is poor BER channel), large interpretational interval is required. Robust Header Compression model relates the interpretation interval with out of synchronization probability as [20]:

$$Po_{oS} = \frac{\in}{L_B}(1 - \frac{1}{L_B})^W IRT \quad (3)$$

where;

Po_{os} Is the Out of synchronization probability

IRT Is the Intinial Refresh Time out

W Is the Interpretation interval

L_B Is the burst length

$\hat{I}$ Is the average error probability

2.4 Average compression length model

Header compression is possible because there is significant redundancy between header field values within packets, but in particular between consecutive packets belonging to the same flow. ROHC uses correlation of the fields in order to reduce the number of fields transmitted. In the best case, ROHC only needs to send Sequence Number (SN) encoded by W-LSB. It can compress IP headers to be as short as one byte. One of interesting properties of W-LSB is that decompression will not fail unless the number of consecutive packet losses exceeds the Interpretation Interval (II). Hence, II plays an important role in W-LSB and has a direct impact on the performance of ROHC. A small II generates fewer bits and shorter compressed header than a large II and has a faster decompression speed, while a small II might cause a higher decompression failure than a large II. Table 1 shows the relationship between II encoded bits and compressed header size of internet protocol version six packets.

Table 1: Relationship between II, encoded bits and compressed header size [3].

II	Encoded Bits	Compressed header size
0	1	1-byte
1-2	2	1 byte
3-6	3	1-byte
7-14	4	1-byte
15-30	5	3 bytes
31-62	6	3 bytes
63-126	7	3 bytes

The Average Compressed Header length (ACL) measures the average length of IPv6 compressed headers. It is the actual header length of compressed IP packets and is dependent on the compressed header length, initial refresh time out, number of update packets and length of update packet. It is expressed mathematical as [21]:

$$ACL = \frac{(L_1 * (RFI - NUP) + L_2 * NUP)}{RFI} \quad (4)$$

where

L_1 is the compressed header length

RFI is the initialization/refresh time out

NUP is the confident variable

L_2 is the length of the update header

L_2 is the length of the update header

2.5 Packet Loss and Transmiited byte model

ROHC compressor uses a confidence variable *(NUP)* in order to ensure the correct transmission of header information. This means sending the same header packet of each compression level, at least NUP times. The total number of transmitted byte (B_r) in the absence of packet loss is calculated as[3]:

$$B_r = N * \mathrm{NUP}(\mathrm{U}+\mathrm{P}) + \mathrm{N}*(\mathrm{M}-NUP)*(\mathrm{ACL}+\mathrm{P}) \quad (5)$$

When header compression is enabled, single packet losses in effect become burst losses of size :

$$Loss = \frac{C_{k,k+1}}{b} * RTT \quad (6)$$

When G number of packets are lost due to OoS, the decompressor would sent a negative acknowledgement packet and return to its previous state. To regain synchronization, the compressor sends uncompressed update packets to unable the decompressor repair its context table. Therefore, G packet losses in become burst losses of size:

$$Loss_L = G*(\frac{C_{k,k+1}}{b} * RTT) \quad (7)$$

Therefore, the total number of transmitted byte with packet loss -

$\mathrm{Loss_L}$ is calculated as:

$$B_{w/o} = N * \mathrm{NUP}(\mathrm{U}+\mathrm{P}) + \mathrm{N}*(\mathrm{M}-NUP)*(\mathrm{ACL}+\mathrm{P}) + Loss_L*(\mathrm{ACL}+\mathrm{P})$$

where :

N is the number of flows

NUP is the confident variable

U is the uncompressed header size

P is the payload size

ACL is the average compressed header Length

M is the total number of packets

b is the the packet size in bits

RTT is theaverage round - trip time

2.6 Bellman-ford Algorithm

The Bellman–Ford algorithm is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph. The algorithm iterate on the number of arcs in a path. At step k of the algorithm, D(i) records the distance from node i to the destination node through the shortest path that consists at most k arcs and the process is [22]:

Step 1 finds the adjacent node w of the node i such that

$\mathrm{D(w)} + \ell(\mathrm{i,w}) = \mathrm{min_j}(\mathrm{D(j)} + \ell(\mathrm{i,j}))$

where the minimization is performed over all nodes j that are neighbours of node i.

Step 2 updates the distance as

$\mathrm{D(j)} = \mathrm{D(w)} + \ell(\mathrm{j,w})$.

These steps continue to iterate until no changes are made.

2.7 Packet Delay Model

Delay is the elapsed time for a packet to travel from the sender through the network to the receiver.

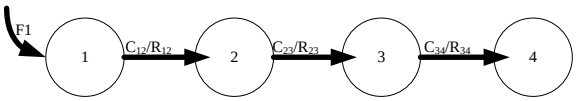

Figure 1: Nodes flow of a Network

Figure 1 illustrates a network of N nodes and flow $f1$. The network is characterized by its transmitting capacity (bps) Ci,j and its propagation delay Ri,j (sec.). If there is no link between nodes i and j, then we have: Ci,j = 0 and Ri,j =∞. A packet transmitted from source to destination is delayed by :

(i) Processing delays

(ii) Transmission delays

(iii) Propagation delays

The delay experienced by packets of flow $f1$ over the link between nodes k and k + 1 is expressed as [23]

$$D_{k,k+1} = W_{k,k+1} + S_{k,k+1}R_{k,k+1} \quad (9)$$

The end-to-end delay experienced by packets of flow $f1$ over their $N-1$ hops from the source to the destination is given as:

$$S_{k,k+1} = \frac{P_{f1}}{C_{k,k+1}} \quad (11)$$

$$D = \sum_{k=1}^{N-1} S_{k,k+1} + \sum_{k=1}^{N-1} R_{k,k+1} + \sum_{k=1}^{N-1} P_{k,k+1} \quad (10)$$

$$R_{k,k+1} = \frac{d}{s} \quad (12)$$

where :

D is the end – to – end delay

$\mathrm{D_{k,k+1}}$ is the delay experienced by packets of flowf1 over the link between nodeskandk + 1

$\mathrm{S_{k,k+1}}$ is the transmission delay

$\mathrm{R_{k,k+1}}$ is the propagation delay

$\mathrm{P_{k,k+1}}$ is the processing delay

$\mathrm{P_{f1}}$ is the packet size

$\mathrm{C_{k,k+1}}$ is the link bandwidth

d is the length of transmitting medium

s is the propagation speed of medium

2.8 Poisson Traffic Model

Teletraffic theory is the application of mathematics to the measurement, modeling, and control of traffic in communication networks. Traffic modeling finds stochastic processes representing the behavior of traffic. Traffic source model is defined from network measures to give an accurate statistical distribution of inter-arrival time and packet size at packet level. It is a mathematical approximation for real traffic behavior. Computer simulations also which pose different requirements for traffic source models. Ideally, a suitable traffic source model should represent real traffic or capture essential characteristics of traffic that have significant impact on network performance. The Poisson arrival process has been a favorite traffic model for data and voice. The traditional assumption of Poisson arrivals is that the aggregation of many independent and identically distributed renewal processes tends to a poisson process. Poisson arrivals with mean rate λ and separated by interarrival times r is express as follows [24]:

$$P_r(\mathrm{Y}=\mathrm{r}) = \mathrm{e}^{-\lambda} \times \frac{\lambda^r}{rfactorial} \qquad (13)$$

where :

λ is Lamba mean rate

r is Numbers of generated packets and inter-arrival time

$e^{-\lambda}$ is the exponential function

3. DEVELOPED SCHEME

Figure 2 shows the flow chart for the improved Intrusion Detection based Secured RObust Header Compression (IDSROHC). Nodes were created using weighted network graph and spatially distributed within a simulation boundary of 100 by 100m using the Random Waypoint Model (RWM) with pause time greater than simulation time. Figure 3 shows the connection matrix used in this research. The work established connection between the nodes using an undirected edge. The row of the connection matrix identifies the source nodes while the column identifies the destination nodes. To establish connection between the nodes, a weight of one was assigned to elements (Cij) of the connection matrix, where Cij is the link between node i and node j.

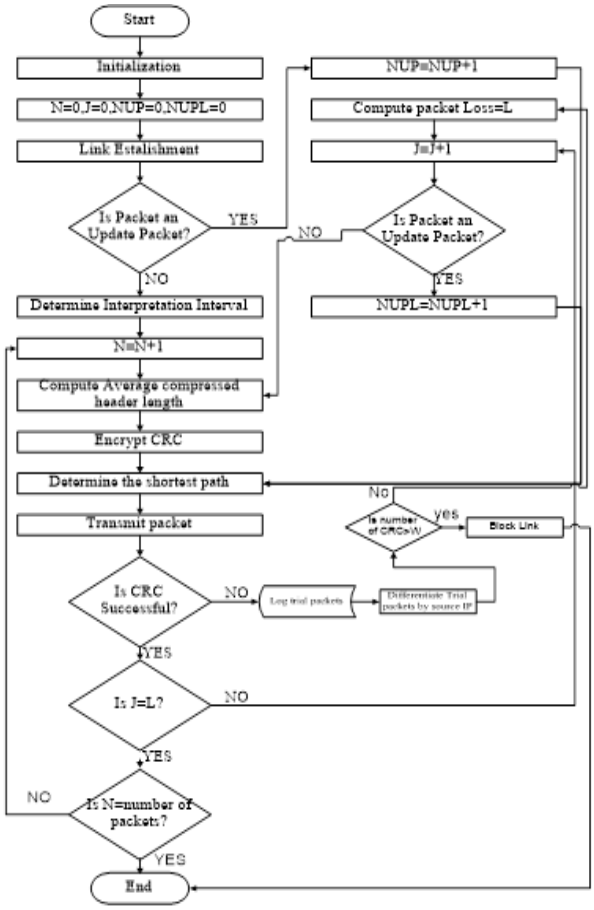

Figure 2: Flowchart for Implementation of IDSROHC

The Bellman-Ford algorithm was use to obtain the shortest path from source node to destination node as shown in Figure 4. Weights were assigned to the edges of the network and stored in sparse matrix. The connection matrix is such that its element C_{ij} is one when there is an edge from vertex i to vertex j and zero when there is no edge. The interpretation interval was obtained using the mathematical expression for robust header compression model. To calculate the average compression length, the obtained value of the interpretation interval (W) was used with Table 1 to obtain the compressed header length. The value of the compressed header length was then use to obtain the average compressed length model using mathematical equation 4. The payload size was calculated using poisson traffic model. The transmitted byte was then calculated. Packets affected by the malicious node were assigned packet size of 0 byte. The uncompressed header was encrypted by applying a 16 bytes block symmetric key encryption. The encrypted uncompressed header was then fed into the CRC algorithm to produce an encrypted CRC.It is assumed that brute force attack is characterized by trial packets of distinct CRC. This is because malicious node try possible combinations to break the encrypted CRC. To detect the malicious node the selective watchdog intrusion detection technique was used. Trail packets were logged and partitioned by destination node. The watchdog intrusion detection system counted the number of distinct CRC of trial packets with the same source node address. It then dropped packets from the nodes with count greater than interpretation interval.

$$\text{Link} = \begin{pmatrix} 0 & \text{if } \mathrm{T} > \mathrm{W} \\ 1 & \text{if } \mathrm{T} < \mathrm{W} \end{pmatrix}$$

Where :

T is the number of trial packets with distinct CRC

W is the interpretation interval

$$C = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & - & - \\ - & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & - & - \\ - & - & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & - & - \\ - & - & - & 0 & 1 & 1 & 0 & 0 & 0 & 1 & - & - \\ - & - & - & - & 0 & 1 & 1 & 0 & 0 & 0 & 1 & - \\ - & - & - & - & - & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 3: Connection matrix

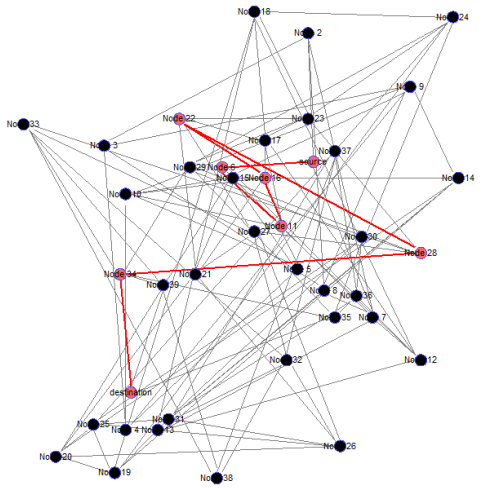

Figure 4: MATLAB Implementation of Bellman ford algorithm

4. PERFORMANCE EVALUATION

In other to evaluate performance, delivery success and throughput were use as describe in mathematical equations 15 and 17 to compare between this developed scheme and that of Secured ROHC [14, 25, 26].

$$\text{Throughput}=\frac{\text{Filesize}}{\text{Transmission\_Time}}(\text{bps}) \quad (14)$$

$$Throughput=\frac{B_r-Loss_L*(\text{ACL}+\text{P})}{D} \quad (15)$$

$$\text{PDS}=(\frac{\text{Total\_number\_of\_packets\_received}}{\text{Total\_number\_of\_packets\_Send}})*100 \quad (16)$$

$$PDS=\frac{B_r}{B_{w/o}}*100 \quad (17)$$

where:

B_r is the total number of packets recieved

$B_{w/o}$ is the total number of packets sent

Figure 5 to 9 show the results obtained in this research as well as the performance comparison between Secured ROHC and IDSROHC technique on the basis of throughput and packet delivery ratio.

4.1 Result of the Effect of Synchronization Probability on Interpretation Interval

Figure 5 shows the effect of Out of Synchronization (OoS) probability on interpretation interval.

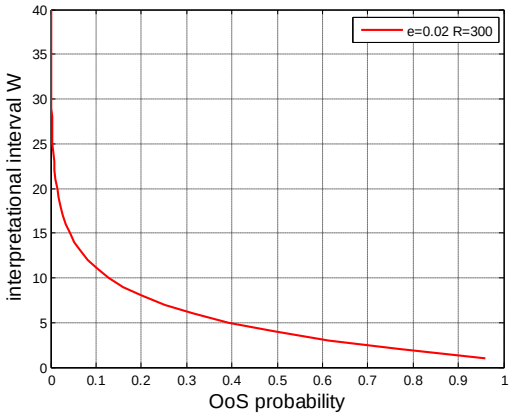

Figure 5: Interpretation Interval against Out of Synchronization (OoS) Probability

It is observed that as the interpretation interval (window based least significant bits) decreases, the OoS probability increases. This implies that an increase in the number of significant bit required for error correction decreases the synchronization probability and vice versa. The result of Figure 5 also shows that with burst length of 5 and average error probability of 2%, the required interpretation interval of 25 is needed to achieve the best compression efficiency and with high robustness.

4.2 Effect of Average Error Probability on Interpretation Interval

Figure 6 shows a graph of interpretation interval against OoS.

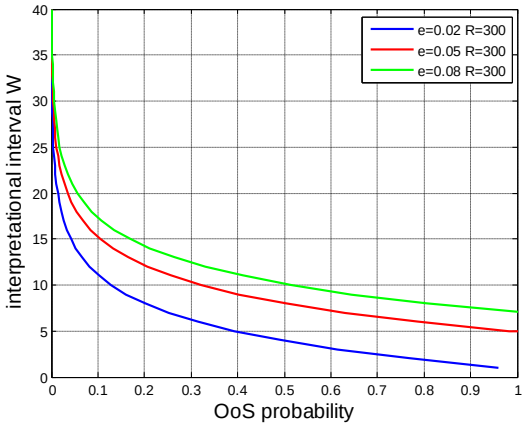

Figure 6: Interpretation Interval against out of OoS

From Figure 6, it is observed that for average error probability of 2%, a required interpretation interval of 25 is obtained. For average error probability of 5%, a required interpretation interval of 28 is obtained. For average error probability of 8%, a required interpretation interval of 30 is obtained. This implies that as the average error probability increases, the interpretation interval (window based least significant bits) needed to correct the error also increases. A burst length of 5 and average error probability of 2% are standard value for stable mobility.

4.2 Result of Relationship between out of Synchronization Probability and the Initial Refresh Timeout

Figure 7 shows the graph of OoS probability against initial refresh timeout for average error probability of 2%, 5% and 8%.

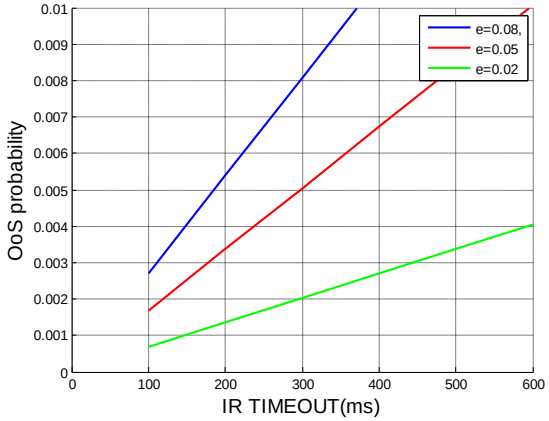

Figure 7: Out of synchronization probability against initial refresh time out

It is observed that with increase in the average error probability, the OoS probability also increases. This implies that increase in average error probability increases the probability of losing synchronization between the compressor and decompressor. Initial refresh timeout ensures context synchronization between the compressor and decompressor. A low initial refresh timeout means increase in robustness due to frequent updating of context table. Figure 7 shows that decrease in initial refresh timeout increases the OoS probability. Initial refresh timeout which result to OoS of 10% of average error probability is recommended to be used for design of Robust header compression system [27]. In this work an initial refresh timeout of 300ms and average error probability of 2% is used.

4.3 Comparison of IDSROHC and Secured ROHC Performances using packet delivery success and network throughput.

Figure 8 and 9 shows the comparison in terms of packet delivery success and throughput between Secured ROHC and IDSROHC.

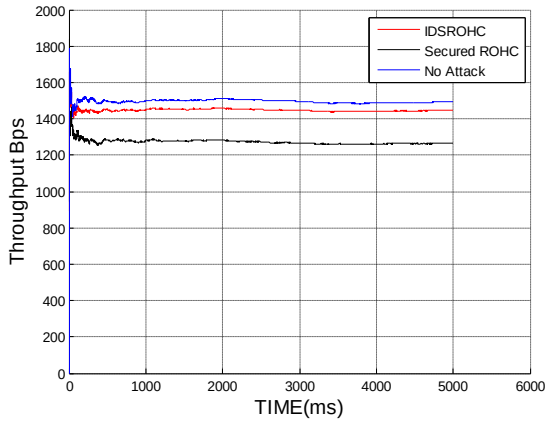

Figure 8: Comparing throughput of Secured ROHC and IDSROHC

Figure 8 shows the comparison of throughput for packets sent without attack and also with attack when the network packets were protected against brute force attack by Secured ROHC and IDSROHC. For the case of Secured ROHC and IDSROHC, the destination node was subjected to brute attack. From the result of Figure 8, the average throughput of network without attack was obtained as 1455 bit per seconds. The average throughput of network for Secured ROHC under brute force was 1367 bit per seconds while that of IDSROHC was 1435 bit per seconds.

The percentage improvement in average throughput for IDSROHC over Secured ROHC was calculated as 4.97%.

Figure 9 shows the result for the comparison of packet delivery success between Secured ROHC and IDSROHC.

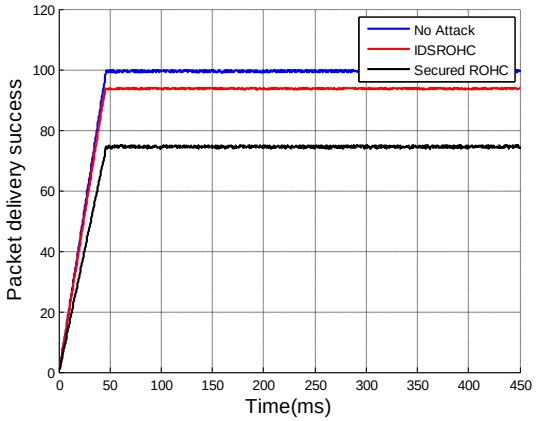

Figure 9: Comparing Packet Delivery Success of Secured ROHC and IDSROHC

The Secured ROHC recorded an average packet delivery success of 72% against brute force attack. This means that 72% percent of packet sent by source was successfully delivered to the destination node. With IDSROHC used to protect packet against brute force attack, 93% average packet delivery success was achieved. This implies that 93% of packet sent by source node was successfully delivered to the destination node. The percentage improvement of IDSROHC over Secured ROHC was calculated to be 29%. The work was simulated using MATLABR2013B as shown in Figure 10 and the simulation parameters used are shown in Table 2.

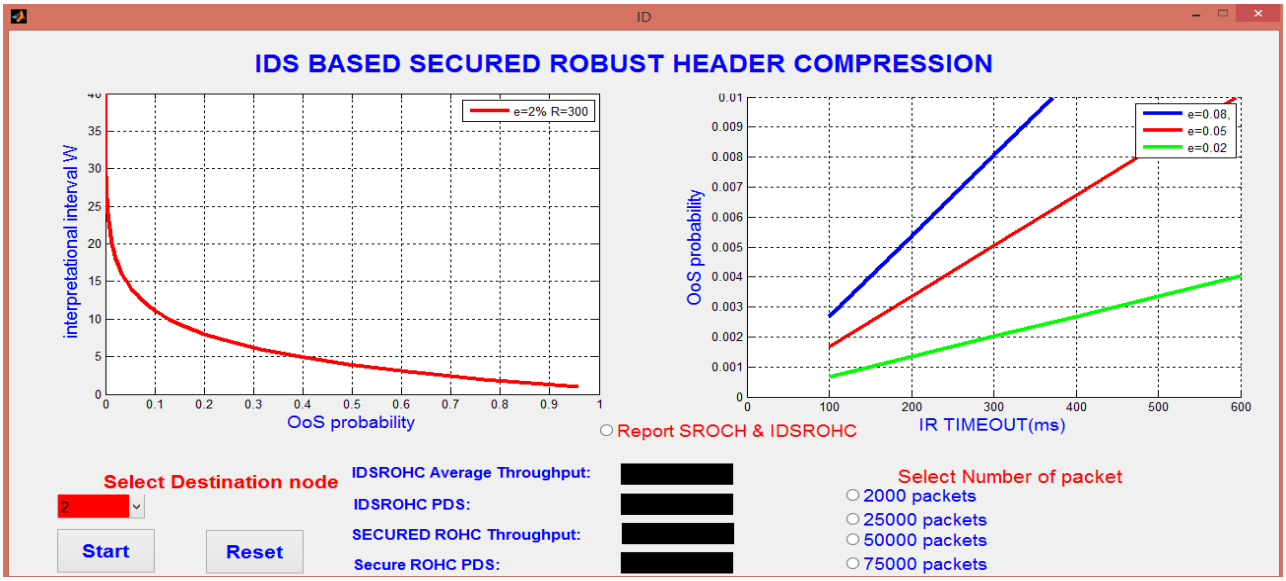

Figure 10: Developed Simulator for Improved Intrusion Detection Secured Robust Header Compression.

Table 2: Simulation Parameters

PARAMETER	VALUE
Simulator	Matlab.
Studied protocol	IPv6.
Area	100 by 100 meters
Total number of nodes	40 nodes.
Routing Algorithm	Bellman-ford.
Link Bandwidth	200KBPS
Number of packets (M)	75000 packets
Uncompressed packet header size	60 byte
Mobility Model	Random way point.
Simulation Time	500seconds.
Confidence variable	4
Area	100 x 100m.
Keylength	16 byte
Block length	16byte
Burst length L_B	5
average error probability (Ɛ)	2%
Traffic Model	Poisson traffic Model
Length of update packet	60 octet
Initial Refresh Time out	300 seconds
Payload size Lamda	230 bytes
Processing time Lamda	2ms

5. CONCLUSION

In other to mitigate the brute force attack associated with Secured ROHC, an improved Intrusion Detection Secured RObust Header Compression (IDSROHC) technique has been developed using watchdog based intrusion detection system. This was developed on a MATLAB graphical user interface platform. The result obtained showed that when seventy five thousand packets were transmitted from the source node to destination node with IDSROHC use against brute force attack, throughput and packet delivery success improvement of 4.97% and 29% were recorded over Secured ROHC. Further study can be done to implement these security features into a kernel of ROHC and evaluating in an emulate environment with real systems.

6. ACKNOWLEDGEMENTS

The authors are grateful to Prof M.B. Mu'azu, Dr. E.A. Adedokun and the Computer and Control research group for their teachings, technical and professional advice in the course of this work.

7. REFERENCES

[1] Cheng, B.-N. and S. Moore, *Securing RObust Header Compression (ROHC).* IEEE Military Communications Conference, MILCOM 2013-2013 2013: p. 1383-1390.

[2] Majanen, M., P. Koskela, and M. Valta, *ConstrainedApplicationProtocolProfileforRobustHeaderCompressionFramework*, in *The Fifth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*. 2015. p. 47-53.

[3] Cha, H., et al. *Improving packet header compression with adaptive sliding window size*. in *2015 International Conference on Information Networking (ICOIN)*. 2015. IEEE.

[4] Chishti, M.A. and A.H. Mir, *Survey of Header Compression Techniques over Multiprotocol Label Switching (MPLS).* Int. J. Com. Dig. Sys, 2015. **4**(2): p. 121-136.

[5] Bow-Nan, C., et al., *MANET IP Header Compression.* MILCOM 2013-2013 IEEE Military Communicatinos Conference, 2013: p. 494-503.

[6] Sandlund, K., G. Pelletier, and L. Jonsson, *The RObust Header Compression (ROHC) Framework Network Working Group Request for Comments: 4996*. 2010. p. 1-94.

[7] Gavaskar, S., E. Ramaraj, and R. Surendiran, *A Compressed Anti IP Spoofing Mechanism using Cryptography*. International Journal of Computer Science and Network Security, 2012. **12**(11): p. 137-140.

[8] Can, O. and O.K. Sahingoz, *A survey of intrusion detection systems in wireless sensor networks.* 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), 2015, 2015: p. 1-6.

[9] Dua, D. and A. Mishra, *Selective Watchdog Technique for Intrusion Detection in Mobile Ad-Hoc Network.* International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks (GRAPH-HOC) 2014. **6**: p. 1-12.

[10] Rawat, P. and J.-M. Bonnin. *Designing a header compression mechanism for efficient use of IP tunneling in wireless networks*. in *7th IEEE Consumer Communications and Networking Conference (CCNC)*. 2010. IEEE.

[11] Shivare, M.R., Y.P. S.Maravi, and S. Sharma, *Analysis of Header Compression Techniques for Networks: A Review*. International Journal of Computer Applications, 2013. **80**: p. pp 14-20.

[12] Jivorasetkul, S., M. Shimamura, and K. Iida. *End-to-end header compression over software-defined networks: A low latency network architecture*. in *4th International Conference on Intelligent Networking and Collaborative Systems (INCoS)* 2012. IEEE.

[13] Madsen, T.K., et al. *Design and evaluation of ip header compression for cellular-controlled p2p networks*. in *IEEE International Conference on Communications, 2007. ICC'07.* 2007. IEEE.

[14] Mohamad, I.J., *End-to-End QoS Improvement using IPv6 Header Reduction over MPLS.* International Journal of Computer Applications, 2013. **80**: p. pp27-32.

[15] Hermenier, R., F. Rossetto, and M. Berioli. *A simple analytical model for RObust Header Compression in correlated wireless links*. in *8th International Symposium on Wireless Communication Systems (ISWCS), 2011* 2011. IEEE.

[16] Li, R., et al., *Method and device for decoding by using window-based least significant bits in robust header compression* U.S. Patent No. 8,418,037. Washington, DC: U.S. Patent and Trademark Office., 2013.

[17] Cheng, B.-N., J. Wheeler, and B. Hung, *Internet protocol header compression technology and its applicability on the*

tactical edge. Communications Magazine, IEEE, 2013. **51**(10): p. 58-65.

[18] Batabyal, S. and P. Bhaumik, *Mobility models, traces and impact of mobility on opportunistic routing algorithms: A survey*. IEEE Communications Surveys & Tutorials, 2015. **17**(3): p. 1679-1707.

[19] Pramanik, A., et al. *Simulative study of random waypoint mobility model for mobile ad hoc networks*. in *Communication Technologies (GCCT), 2015 Global Conference on*. 2015. IEEE.

[20] Hermenier, R., F. Rossetto, and M. Berioli, *On the Behavior of RObust Header Compression U-mode in Channels with Memory*. IEEE Transactions on Wireless Communications, 2013. **12**(8): p. 3722-3732.

[21] Wang, B., et al. *On implementation and improvement of robust header compression in UMTS*. in *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*. 2011. IEEE.

[22] Osunade, O., *A Packet Routing Model for Computer Networks.* International Journal of Computer Network and Information Security, 2012. **4**(4): p. 13.

[23] Salehin, K.M., et al., *Scheme to Measure Packet Processing Time of a Remote Host through Estimation of End-Link Capacity*. IEEE TRANSACTIONS ON COMPUTERS, 2013. **Vol. X**: p. pp 1-14.

[24] Zhang, J., et al. *A survey of network traffic generation*. in *Third International Conference on Cyberspace Technology (CCT 2015)*. 2015. IET.

[25] Meenakshi, *Impact of Network Size on Performance of Wireless Network Topology*. International Journal of Advance Research in Computer Science and Management Studies, 2014. **2**(9): p. 175-179.

[26] Chauhan, D. and S. Sharma, *Addressing the bandwidth issue in end-to-to header compression over ipv6 tunneling mechanism.* I.J.Computer and information security, 2015. **9**: p. 39-45.

[27] Hermenier, R., F. Rossetto, and M. Berioli, *On the Behavior of RObust Header Compression U-mode in Channels with Memory*. Wireless Communications, IEEE Transactions on, 2013. **12**(8): p. 3722-3732.