JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(2), JUNE, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

# Development of a Secured Robust Header Compression Scheme using Cipher Based Message Authentication Code to Mitigate False Initialization/Refresh Attack

A. A. Isah, M. B. Muazu, E. A. Adedokun, I. J. Umoh, O. W. Salami,
*Department of Computer Engineering,*
*Ahmadu Bello University Zaria,*

**ABSTRACT**

*This research presents a secured robust header compression using Cipher based message authentication code to mitigate false initialization/refresh attack (CMACROHC). Several research works were done to address false initialization refresh attack (FIA) mostly focusing on intrusion detection system and ROHC network performance while neglecting the data authentication and integrity aspect of the ROHC. A cipher base message authentication code (CMAC) was introduced and integrated into ROHC to improve data authentication and integrity. In this work The CMAC technique was used to calculate and generate an authentication tag of the uncompressed packet using a secret symmetric key at the compressor and transmit the packet along with the tag to the decompressor. The tag is recalculated by the decompressor and compared with the received tag from the compressor to verify if both tags match. As both tags match, it shows that the uncompressed packet is authentic, from a legitimate source (sender) and indicates that the data has not been compromised. The performance of The CMACROHC was compared with IDSROHC to determine how well each of the schemes can protect and secure the packet along the broadcast channel. The developed scheme was simulated in network simulator version 3 (NS3). Its performance was evaluated using throughput and packet delivery success.*

**ARTICLE INFO**
*Article History*
Received: October, 2023
Received in revised form: December, 2023
Accepted: February, 2024
Published online: June, 2024

**KEYWORDS**
Secure Robust Header Compression Scheme, Chipper based Message Authentication Code

## INTRODUCTION

Rapid expansion of users of the internet and mobile application has led to high demand of limited allocated Radio Frequency (RF) spectrum in recent years. In addition, with the desire to migrate to internet protocol version 6 (IPv6) (or dual stack IPv4/IPv6) in which Internet Protocol (IP) header increase drastically in size, exhausting sizeable part of RF. It is paramount to decrease IP header overhead (4) because in wireless networks, bandwidth preservation is of high importance (1). The use of Window Least Significant Bit algorithm (W-LSB) couple with feedback makes the ROHC robust against link bit error rate that also result in less long round trip time and obtain high compression gain. The stateful nature of the compression scheme expose it to the potential risk of security breach that applying authentication and encryption can mitigate the risk (1). The ROHC protocols is design to be robust over different link technology with different quality by providing efficient and robust compression header (16). It usually set up a similar CID at the source (compressor) and destination (decompressor) via sending the uncompressed full packet, then increment to higher compression level utilizing different encoding techniques and packet format (17).

The usual ROHC operation is that an uncompressed packet will be transmitted by the compressor along with an extra initialization/refresh (IR) headers for context establishment (18), Then compressor transmit subsequent packets compressed when it is certain the decompressor has the context. When fields of the packet change, compressor may alter context bit by

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

transmitting extra IR header, then the packet will be decompress by the decompressor and perform CRC check, while acknowledgment (ACK) will be sent by decompressor to that effect for a successful decompressed packet (3). Decompression may fail due to an out of sync of context cause by cyclic redundancy check (CRC) failure and a Negative ACK (NACK) for packet not successfully decompress. When compressor gets NACK, it then try to repair the context damage by sending a Context Repair header or IR (4). Only when decompressor has adequate stored information to map the headers to a particular CID that Successful decompression occur with any mismatches resulting in errors (19).

## LITERATURE REVIEW

[4] Presented solutions to mitigate the attacks identified by the work that were associated with ROHC. The attacks were (a) False initial/refresh attack. (b) False acknowledgement attack. (c) False negative acknowledgement attack. While the solutions were IPsec and encryption along with CRC. Experimental results showed improvement in network load, packet delivery success and out of synchronization under the attacks. However, the IPsec solution was very expensive to practically implement because of the large overhead associated with the IP tunneling, while the CRC is 3-8 bits, this limited the possible combination to only 256. A malicious node can still attempt to use brute force attack by transmitting fake packet with possible combination, which may also lead to denial-of-service attack.

[7] Presented an Intrusion Detection System based Robust Header Compression (IDSROHC) using secured watchdog approach. Trial packets were logged and partitioned by destination node after the CRC failure, the number of distinct CRC of the trial packet with same source address were counted and packet from the node with count greater than interpretation interval were dropped. The approach worked well in terms of monitoring but may not be good in terms of identifying the real attacker (misbehaving nodes may not be always an attack node) because setting up the interpretation interval may affect the process due to CRC bits of only 3-8 bits. Also, in case of collaborative attack, where two nodes will collaborate and drop a packet without the knowledge of the 3rd legitimate node. So, a spoofing attack can be done on different nodes to lure the watchdog to be blocking nodes or links or even the IDS can be bypassed.

[14] reviewed various existing header compression schemes. It identified ROHC as the promising header compression scheme and also introduced 6LowPAN header compression schemes. The 6LowPAN was used to reduce the size of the upper layer header. The 6LowPAN was compared with ROHC and saw that the IP in ROHC can be replaced with 6LowPAN in the TCP stack to alleviate problems that the IP created. Also, the 6LowPAN doesn't include the TCP at the layer 4 because it wasn't needed due to the fact that it has nothing to do with packet loss in wireless. However, the 6LowPAN didn't investigate the security aspect of the ROHC which can rendered the work vulnerable to attacks.

(9) aimed to reduce packet header by compression for unidirectional and bidirectional optimistic mode of ROHC in real time video streaming for UDP/IP and HTTP/TCP flow over error free channel. The result showed better compression gain than previous studies. But only focused on compression gain without considering the security aspect which is essential and of paramount importance.

(15) investigate a bi-directional ROHC by integrating reinforcement learning with the introduction of partially observable Markov decision process (POMDP) in which the compressor is the agent while the decompressor, header source and channel make up the environment. Deep Q network (DQN) was adopted that takes the history of actions and observations as input and output Q values of the corresponding actions. The result show good performance in terms of transmission efficiency and addressed scalability drawback in exiting work like that of dynamic programming (DP) in the case of large finite states. The solution suffers computational complexity due to feedback delay and complex channel mode when compared with DP that was proposed in previous works. The work falls short in the event of more complex environment such that include complex security solution like

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

encryption with multi-agent reinforcement learning.

From the literature reviewed, most of the solutions focused on securing the nodes rather than the packets which solved the problem of detection on per node basis. The likes of watchdog approach implemented in IDSROHC only detect the malicious node which is not sufficient to deter an attack on the packet. Others suggested encryption mechanism to encrypt only the CRC. Due to the smaller size of the CRC, a brute force can be performed on it to gain access to the packet. A well secured scheme that can secure the confidentiality as well as the integrity of the packets is needed. Therefore, implementing a scheme that can assure the data integrity, like CMAC, to mitigate FIA attack is the basis for the motivation of this research.

**Robust header compression (ROHC)**

The ROHC usually set up a similar CID at the source (compressor) and destination (decompressor) via sending the uncompressed packet, then increment to higher compression level utilizing various encoding techniques and packet format(10). The ROHC framework incorporates a compressor at the source end to compress the packet and a decompressor at the destination end for returning (decompression) the packet back to its original format and size. The channel where the compression take place is termed as ROHC channel. The ROHC uses context to store bits of data in the decompressor and compressor (7). As can be seen in Figure 2.1, CID mapped some static field information and requires to be synchronized between decompressor and compressor for every packet flow and hop (4). Effective decompression possibly happens as decompressor has sufficient stored data to map the header to a unique CID, with every alteration bringing about invalidation (19).

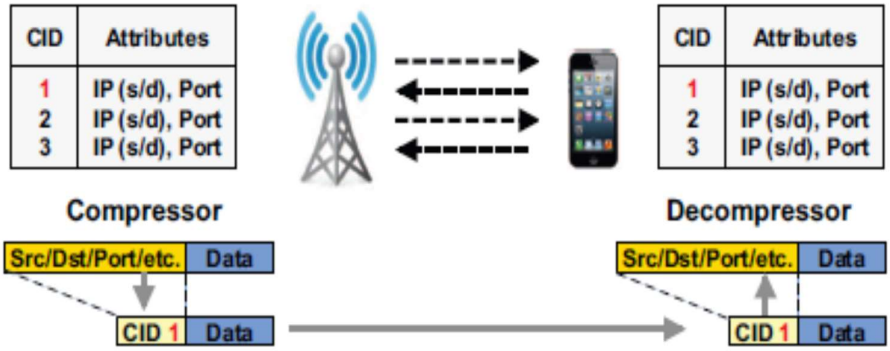



**Figure 1:** ROHC(4)

Robust Header Compression model relates the interpretation interval with out of synchronization probability as express in equation 2.1(12)

$$Po_{oS} = \frac{\epsilon}{L_B}(1 - \frac{1}{L_B})^W IRT \qquad (2.1)$$

where

$Po_{oS}$ is the Out-of-synchronization probability
IRT is the Initial Refresh Time out
W is the Interpretation interval
$L_B$ is the burstlength
$\epsilon$ is the average error probability

The W-LSB is defined by an interpretation interval. Interpretation interval is the maximum number of packets that are lost in a row without losing context synchronization. It is expressed mathematically as follows (12):

$$W = [-o_f, 2^b - 1 - o_f] \qquad (2.2)$$

where:
W is the interpretation interval
$o_f$ is the offset.
b is the least significant bit

**ROHC false IR attack (FIA)**

In FIA, the malicious nodes overheard the IR headers for CIDs set up. Whenever it heard the particular context, it will first spoof the compressor (the sender) and transmit another IR to receiver (decompressor) adjusting the CID in a slight manner as shown in Figure 2.5.

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(2), JUNE, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

Therefore, subsequent compressed packets got at the decompressor end will fail CRC and will be discarded. However, the malicious node knowing the genuine CID, most likely will decompress the packet effectively (Cheng & Moore, 2013).

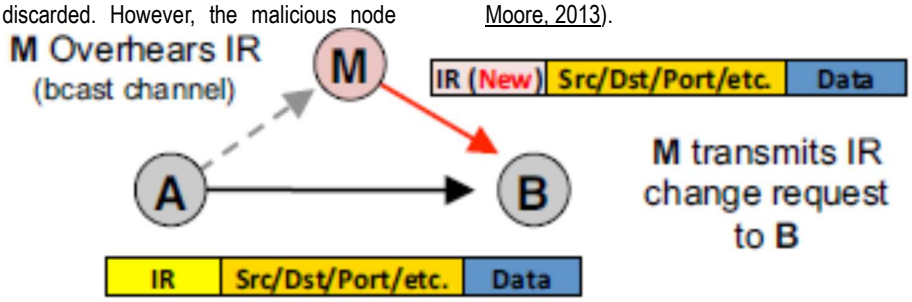



**Figure 2:** ROHC False IR Attacks (FIA) (4).

In Figure 1.2, legitimate Node A transmits an IR packet meant for legitimate node B. Due to broadcast channel of transmission used, malicious node M eavesdrop the IR packet and be able to alter the IR request to B, causing B to be unable to decompress the packet headers as compressed from A. However, the malicious node M can in any case get and decompress packets gotten from A because of its knowledge of the CID mapping. Under the FIA Attack, when the decompressor isn't designed to send feedback, at that point this state will last throughout.

The decompressor will send a NACK to the compressor owing to its disability to decompress the compressed packet sent when the setup is configured to send feedback. But the malicious node having the IR knowledge will be able to decompress all the packet because it has the context set up (CID). On getting NACK, the compressor in an attempt to repair the CID will send a new IR, but the malicious node been in the channel will again capture and alter it before sending to the decompressor, this situation will be repeated over and over, while the end result is that only fraction of the packet will be gotten by the decompressor, depending on the uncompressed frames sent by the compressor before moving to the compression stage (compress packet). The network load will increase due to the large amount of uncompressed packet along with IR, negating the benefit of utilizing ROHC (4).

**Cipher-base message authentication code (CMAC)**

CMAC is a MAC algorithm built upon a block cipher with symmetric key. It can be considered a mode of operation of the block cipher and suitable for data that an accepted block cipher is more available than a hash function. The algorithm is a modification of Cipher Block Chaining (CBC-MAC) proposed by Black and Rogaway and analyzed under the name Extended Cipher Block Chaining (XCBC), which successfully tackles the security issues of CBC-MAC. It also provides some level of guarantee for integrity of information than error detecting code or checksum, because the two are used to detect accidental data modifications only, while CMAC is used to detect deliberate, unauthorized, as well as accidental alterations of data (11).

CMAC uses AES as a building block where it takes a variable length message, the message length and a secret key in octets as an input and gives output of a fixed-bit string called a MAC. The key for CMAC algorithm is the block cipher key and depends on the choice of an underlying block cipher (choosing AES in this case). Also, the design of the block cipher functions as the forward transformation and the other as the reverse transformation (as in the AES algorithm) influences the choice of the block cipher. The CMAC mode only employ the forward and discard the reverse function.

The forward cipher function is a variation on bit strings of a block denoted as CIPH*K* and the block denoted by *b*, while the block length called the block size is *b* = 128 in the case of AES algorithm and the key is denoted by *K*.

Pseudocode for CMAC Generation and Verification:

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

Notations

b Block bit length.
Rb Cipher constant string for subkey generation with block size b.
K The AES cipher key.
K1 1st subkey.
K2 2nd subkey.
M Data message.
Mi The ith block of the formatted message.
Mn * the last block (possibly a partial block) of the formatted message.
Mlen bit length of the message.
n the formatted message number of blocks.
T MAC tag.
Tlen MAC tag bit length. (8)

Generation of Subkey:

The CMAC process of subkey generation:

*Prerequisites*:
block cipher CIPH with block size *b*;
block cipher key *K*.
*Output*:
subkeys *K1*, *K2*.
*Suggested Notation*:
SUBK(*K*).
*Steps*:
1. Let L = $CIPH_k(0^b)$.
2. If MSB1(L) = 0, then K1 = L << 1;
Else K1 = (L << 1) ⊕Rb;
3. If MSB1(K1) = 0, then K2 = K1 << 1;
Else K2 = (K1 << 1) ⊕Rb.
4. Return K1, K2. (8)

The first Stage is application of block cipher to the entire '0' bits block. The second stage shows the derivation of the first subkey from the string output by a one-bit left shift, followed by conditionally XORing a constant based upon the block size. While third stage shows the derivation of the second subkey same way as the first subkey.

Generation of MAC:

The process of CMAC tag generation:

*Prerequisites*:
Cipher block CIPH with block size *b*;
MAC key *K*;
Length parameter of MAC *Tlen*.
*Input*:
bit length of message *M, Mlen.*
*Output*:
MAC *T* of bit length *Tlen.*
*Suggested Notation*:
CMAC (M, *K*, *Tlen*) or if *Tlen* is understood from the context, then CMAC(*K*, *M*).
*Steps*:
Apply the subkey generation process to *K* to output *K1* and *K2*.
If *Mlen* = 0, let *n* = 1; else, let *n* = (*Mlen*/*b*).
Let *M*1, *M*2, ... , *Mn*-1, *Mn* * denote the unique sequence of bit strings such that *M* = *M*1 || *M*2 || ... || *Mn*-1 || *Mn* *, where *M*1, *M*2,..., *Mn*-1 are complete blocks
If *Mn* * is a complete block, let *Mn* = *K1* ⊕ *Mn* *; else, let *Mn* = *K2* ⊕ (*Mn* *||$10^j$), where *j* = *nb*-*Mlen*-1.
Let *C*0 = $0^b$ .
For *i* = 1 to *n*, let *Ci* = CIPH*K*(*Ci*-1 ⊕ *Mi*).
Let *T* = MSB*Tlen*(*Cn*).
Return *T*. *(8)*

First stage shows the generation of subkeys from the main key. In the second to fourth stage, the inputted message is arranged into complete blocks sequence in which the last block has been XORed by a subkey. Two (2) cases emerge:

The message will be partitioned into complete blocks, if length of the message is a positive multiple of block size. The first subkey will then be masked with the final block, the last block after partitioning the message is interchanged with the XOR of the last block with the first subkey. The output blocks sequence is the formatted message.

Then, the message will be partitioned into complete possible blocks, to a complete blocks sequence followed by a last bit string that its length is less than the block size when the length of the message is not a positive multiple of the block size. Whereas, the final bit string will be appended with a padded string, a single '1' bit followed by the number of minimum '0' bits, possibly none that are necessary to form a complete block. The complete last block is interchanged with the second sub key. The output blocks sequence is the formatted message.

In 5 to 6, CBC with the 0 block being the initialization vector, will be applied to the formatted message. While in Stage 7 to 8, the resulting CBC output blocks are truncated according to the MAC length parameters that are associated with the key, and the output is returned as the MAC tag. Note that the formatting of the entire message (3 to 4) is not necessary to be completed prior to the cipher

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(2), JUNE, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

block chaining Stage 5 to 6. In that case, the iterations of Stage 5 may be executed "on the fly," i.e., on each successive block of the message as soon as they are available for processing. Stage 4 may be delayed until the final bit string in the partition is available; the appropriate case and value of j, if necessary, can be determined from the length of the final bit string. In such an implementation, the determination in Stage 2 of the total number of blocks in the formatted message may be omitted, assuming that the implementation has another way to identify the final string in the partition as shown in Figure 2.9.

Alternatively, the sub keys may be pre-computed and stored along with the key as the algorithm inputs.

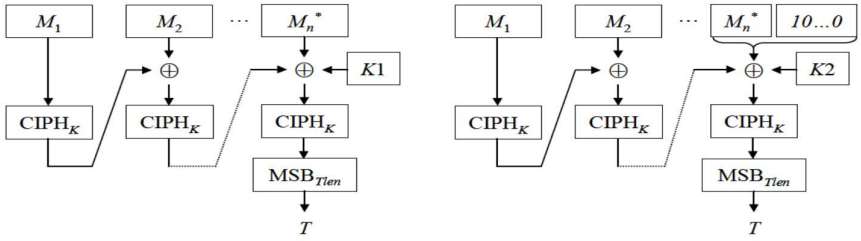



**Figure 3:** MAC Generation for the Two Scenario (8)

Verifications of MAC:
The process of MAC verification with CMAC:
Prerequisites:
Block cipher CIPH with block size b;
Block cipher key K;
Sub keys K1, K2;
MAC length Tlen.
Input:
bit length of message M, Mlen;
received MAC tag T'.
Output:
VALID or INVALID.
Suggested Notation:
VER(K, M, T').
Steps:
1. Applying the process of MAC generation for M to produce T.
2. If T = T', return VALID; else, return INVALID
(8)

In Stage 1, it shows the application of MAC generation process to the message while in Stage 2, the resulting MAC is compared with the received MAC to determine its validity

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

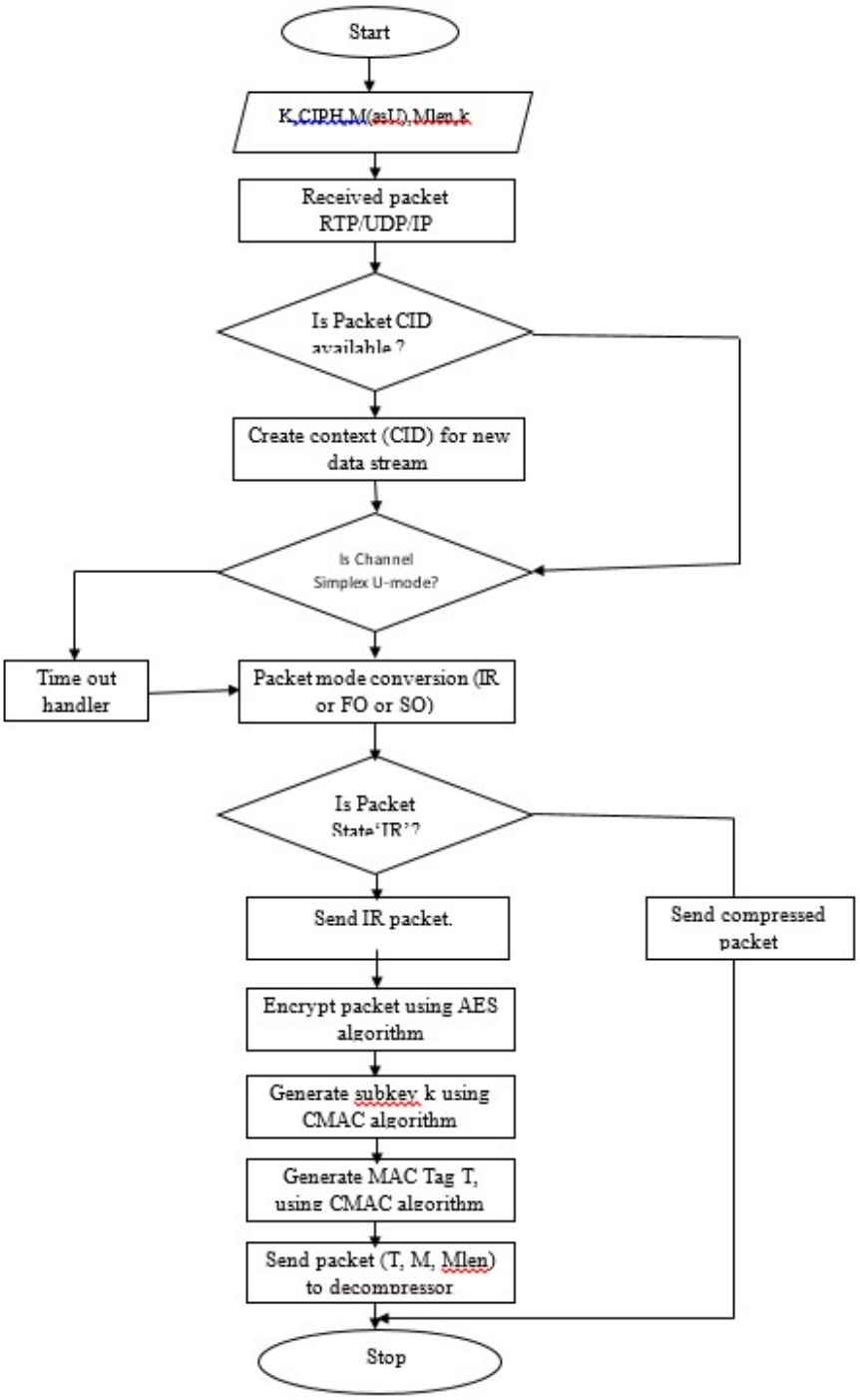



**Figure 4:** The Flowchart of the Developed Work at the Compressor

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

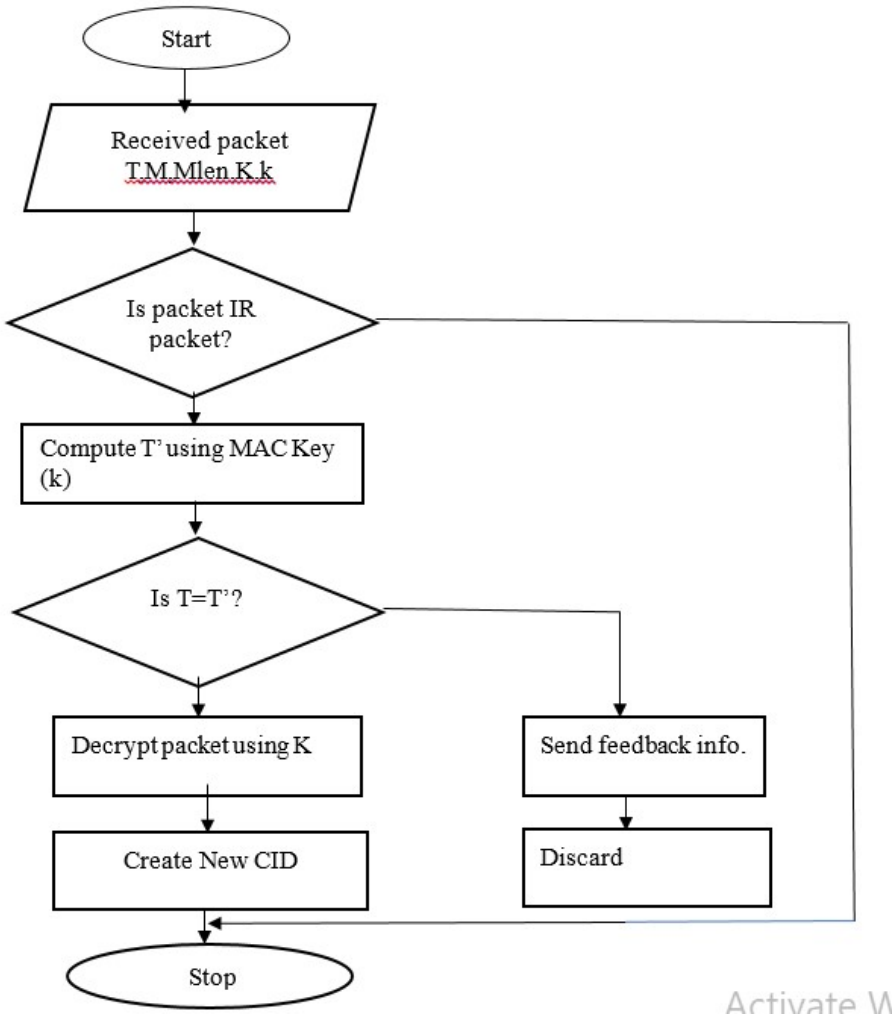



**Figure 5:** The Flowchart of the Developed work at the Decompressor

Where:
U=Uncompressed packet
AES Key=K
MAC Key=k
SN=sequence number
CIPH=Cipher text
T=Mac output or tag

The flowchart shows the step-by-step process of this work. The uncompressed packet (M) was encrypted using AES algorithms and the CMAC takes the AES key K, ciphertext CIPH and generate a subkey (k). It then uses the subkey generated (k) with message (M), message length (Mlen), the cipher text (CIPH) from the AES and generate the CMAC tag T as output. It takes the tag T, the cipher text (CIPH), the message (M) and transmit it to the decompressor at the receiving end. The receiver (decompressor) will recompute the CMAC tag T' and compare with the sent CMAC tag T, if is the same then the message is authentic and has not been compromise (integrity). Decrypt the cipher text and check the sequence number to checkmate replay attack.

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(2), JUNE, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

## RESULTS

### Evaluation of throughput of CMACROHC and IDSROHC under no attack

ROHC channel was subjected to no attack to examine the throughput under the CMACROHC and the IDSROHC. As can be seen in Figure 4.1, the CMACROHC presented a slightly higher throughput of 1460.19bps against IDSROHC of 1391.79bps resulting in 4.91% on average based.

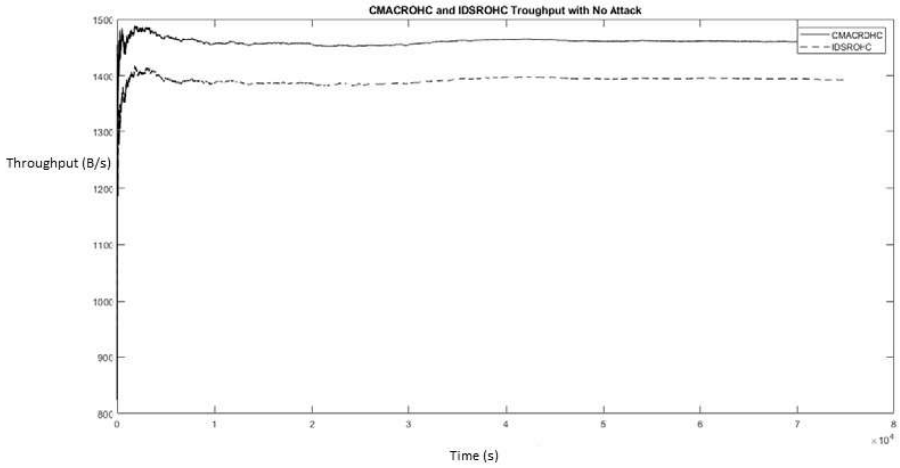



**Figure 6:** Result of Throughput of CMACROHC and IDSROHC under No Attack

This was due to the fact that the IDSROHC used watchdog IDS which monitor each and every node and this will result in delaying packets along the communication channel, because all the nodes must be in active mode to ensure that the current node monitor its next neighbor and verify that the next neighbor transmit the packet to its next neighbor also while the current node send acknowledgement to its previous neighbor. This activity will take time and may lead to low throughput.

### Evaluation of throughput for CMACROHC and IDSROHC under FIA

The throughput of the two schemes were also compared when an attacker-initiated FIA attack on the channel. The CMACROHC secured the channel more than the IDSROHC due to the fact that the packet will be encrypted first with an encryption key and also apply authentication technique (MAC) to increase the level of security before sending it to the decompressor. IDSROHC only focused on the neighbor node and relied solely on it for feedback (positive or negative). In IDSROHC the attacking node may come in between legitimate nodes and deceive the nodes by giving false feedback information. So as can be seen in Figure 4.2, CMACROHC with 1448.70 bps perform better than the IDSROHC with 1258.32 bps on average resulting in 15.13% improvement.

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

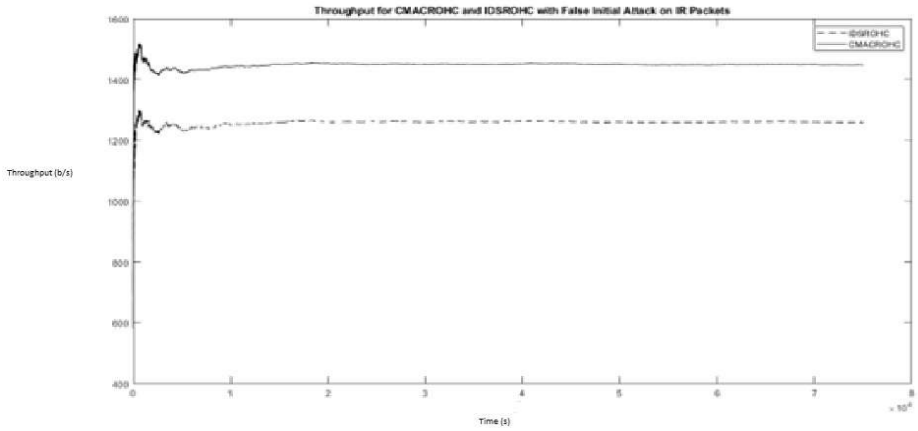



**Figure 7:** Result of Throughput of CMACROHC and IDSROHC under FIA

The 15.13% achievement can be attributed to the fact that CMACROHC has high level of security such that even when the intruder comes into the communication channel and eavesdrop it can only see the packet but cannot alter it, because it doesn't have the decryption key and cannot authenticate the packet. So, the time it will take him to be trying to decrypt the packet and authenticate itself, by then the decompressor has already decrypted and authenticated the packet then move to next step. While for the IDSROHC the process of the watch dog is per node aspect, from first node to the last node in the topology and this will reduce the throughput with security level also decreased.

**Evaluation of CMACROHC and IDSROHC under brute force (NACK) attack**

The two schemes were evaluated under brute force attack in Figure 4.3, where the attacker will try to send a NACK/BF to the compressor to lure the compressor to send a new IR packet whereas the receiver (decompressor) was able to decompress the packet successfully and send an ACK reply. In this case the ACK will override the NACK and the compressor will continue in the SO stage instead of the IR stage. So, the CMACROHC 1444.09bps also outperform the IDSROHC 1331.06bps leading to 8.49%.

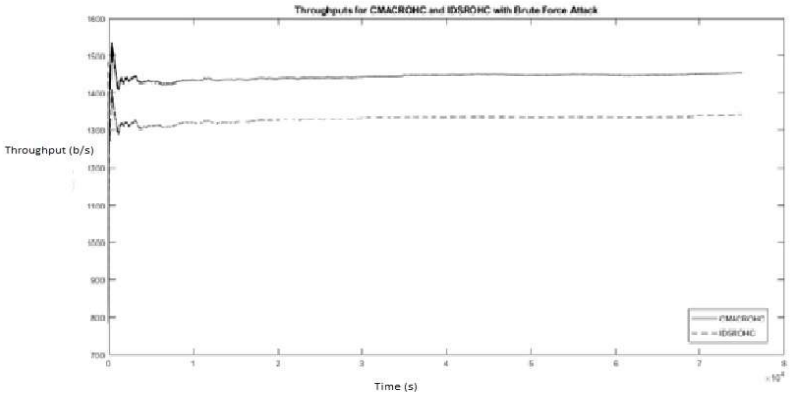



**Figure 8:** Result of Throughput of CMACROHC and IDSROHC under NACK/BF Attack

**Evaluation of packet delivery success of CMACROHC and IDSROHC under no attack**

In the packet delivery success scenario, the two schemes were subjected to no

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

attack to see the performance. The CMACROHC with bandwidth of 92.64bps outperform the IDSROHC with 80.89bps resulting to 14.53% improvement based on Equation (3.1) as shown in Figure 4.4.

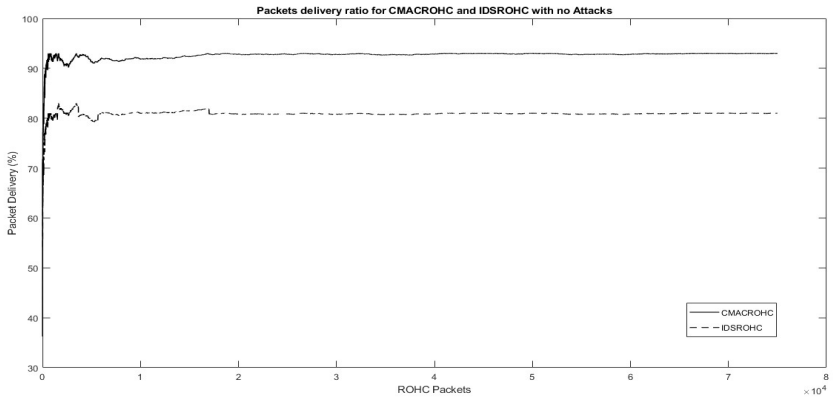



**Figure 9:** Result of Packet Delivery Success of CMACROHC and IDSROHC under No Attack.

The performance improvement of the CMACROHC was observed on the no attack scenario by 14.53% on average because the CMACROHC properly secured the packets along the channel by adding the authentication algorithm to make sure that the packet is authenticated before decompressing it as against the IDSROHC that only focused on the nodes in the network which some nodes can be compromised by the attacker or even act as the attackers.

**Evaluation of Packet Delivery Success of CMACROHC AND IDSROHC under FIA attack**

As shown in Figure 4.5, there was 9.20% improvement for CMACROHC with bandwidth of 86.75bps as compared to IDSROHC with 79.44bps under FIA based on Equation (3.1). This was because the developed scheme has the ability to secure the packet and make it almost impossible for the attacker to get the packet and compromise it. So, during that time the attacker is trying ways to alter the packet, the receiver (decompressor) might have already decompressed the packet and sent an ACK, leading the compressor to move to the compression state and sent all other packet as compressed.

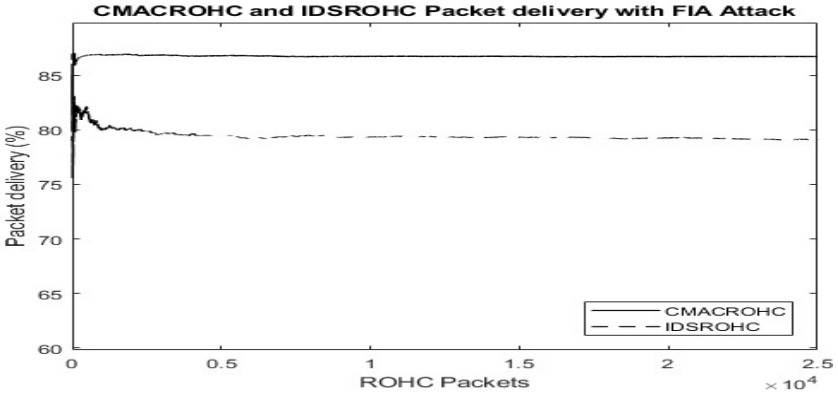



**Figure 10:** Result of Packet Delivery Success of CMACROHC and IDSROHC under FIA

This will make more packets to be received and decompressed as against the IDSROHC that will be monitoring each node, which would give the attacker more time to

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

capture the packet, alter it and send it to the decompressor, thereby reducing the number of packets delivered to the decompressor leading to low packet delivery success rate.

**Evaluation of CMACROHC with IDSROHC under NACK/BF attack**

Figure 4.6 showed the comparison of packet delivery success for CMACROHC and IDSROHC under NACK attack and from the result of the average packet delivery success on the network there was 17.00% improvement for CMACROHC with 80.04bps against the IDSROHC with 68.41bps of bandwidth on average.

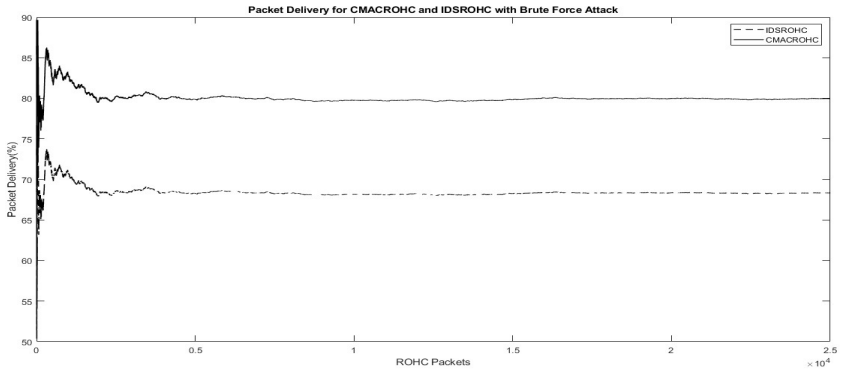



**Figure 11:** Result of Packet Delivery Success of CMACROHC and IDSROHC under NACK/BF.

The average packet delivery success for CMACROHC was obtained to be higher than IDSROHC under NACK attack because the time it will take the attacker to get the packet, study it and send a NACK to the compressor will be higher than the time it will take the receiver (decompressor) to decompress the packet and send an ACK which will override the attackers NACK that may delayed the feedback. Also, in the case of IDSROHC the attacker's node can act as a legitimate node, capture the packet, alter it and send a NACK to the compressor while in the real sense it is dropping the packet from reaching the decompressor which will in turn reduce the packet delivery success.

**Quantitative Comparison of Results**

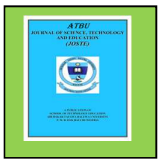
Quantitative comparison is done by comparing the results of the developed scheme CMACROHC, with that of the benchmarked scheme IDSROHC, for throughput and packet delivery success considering the three experimental scenarios of No attack, FIA attack and NACK attack. The comparison was used to determine the efficiency of CMACROHC.

Table 4.3 showed that on No attack test CMACROHC throughput outperformed that of IDSROHC by 4.91% on the average, while it was 14.53% improvement on Packet Delivery Success. On FIA attack, the CMACROHC saw an improvement of 15.13% on throughput and 9.20% on Packet Delivery Success more than IDSROHC. Lastly, on NACK attack the CMACROHC outperformed IDSROHC by 8.49% throughput and 17.00% Packet Delivery Success.

Table 1: Comparison of CMACROHC and IDSROHC Performances with Throughput and Packet Delivery Success.

| **Test** | **ROHC No Attack** | | **ROHC FIA Attack** | | **ROHC NACK Attack** | |
|---|---|---|---|---|---|---|
| | Throughput (bps) | Packet Delivery Success (bps) | Throughput (bps) | Packet Delivery Success (bps) | Throughput (bps) | Packet Delivery Success (bps) |
| Percentage improvement (%) | 4.91 | 14.53 | 15.13 | 9.20 | 8.49 | 17.00 |

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*



## CONCLUSION

This research developed a mechanism called the CMACROHC scheme to mitigate FIA attack that causes packet dropping on decompressor. The attacker uses FIA to modify IR-packets used to build IR-Table for decompressing ROHC packets and made decompression not possible due to IR-Table mismatch.

CMACROHC was developed by applying the CMAC on the ROHC to authenticate uncompressed initialization/refresh (IR) packet between the compressor and the decompressor to stop an attacker from compromising and altering the packets. The authentication provides an assurance to the receiver that the packet is from the original sender by comparing the computed tag of the receiver with the transmitted tag using secret key shared by both parties.

The work was simulated in NS3 testbed, and the result was compared with that of IDSROHC using packet delivery success and throughput as performance metrics.

Under FIA attacks, CMACROHC yielded 1448.70bps throughput against the IDSROHC that gave 1258.32bps, showing CMACROHC performed better with 15.13%. Also, CMACROHC yielded packet delivery success of 86.75bps, while IDSROHC recorded 79.44bps which is 9.20% lower than CMACROHC in the same FIA attacks tests. In NACK attack tests, CMACROHC produced throughput of 1444.09bps which is 8.49% higher than IDSROHC throughput of 1331.06bps. Lastly, CMACROHC achieved a 17.00% packet delivery success rate better than IDSROHC with its 80.04bps against IDSROHC's 68.41bps

## REFERENCES

1. Chauhan, D., & Sharma, S. (2015a). Addressing the Bandwidth issue in End-to-End Header Compression over IPv6 tunneling Mechanism. International Journal of Computer Network and Information Security, 7(9), 39.
2. Chauhan, D., & Sharma, S. (2015b). Enhancing the Efficiency of IPv6 Tunneling Mechanism by using Header Compression over IPv6 Header. International Journal of Advanced Research in Computer and Communication Engineering, 4(4), 446.
3. Chen, X., Guo, F., Dang, P., & Wu, L. (2012). A survey of ROHC header compression schemes. Proceedings of 2012 2nd International Conference on Computer Science and Network Technology,
4. Cheng, B.-N., & Moore, S. (2013). Securing robust header compression (rohc). Military Communications Conference, MILCOM 2013-2013 IEEE,
5. Cheng, B.-N., Wheeler, J., & Hung, B. (2013). Internet protocol header compression technology and its applicability on the tactical edge. IEEE Communications Magazine, 51(10), 58-65.
6. Cheng, B.-N., Zuena, J., Wheeler, J., Moore, S., & Hung, B. (2013). MANET IP header compression. MILCOM 2013-2013 IEEE Military Communications Conference,
7. Egbugha, M. C., Umoh, I., & Tekanyi, A. (2017). Development of an improved intrusion detection based secured robust header compression technique. International Journal of Computer Applications, 161(7).
8. Dworkin, M. (2005). NIST special publication 800-38B. *NIST special publication*, *800*(38B), 38B.
9. Farouq, D. B., Alarood, A. A., Aljojo, N., & Abubakar, A. (2020). Unidirectional and bidirectional optimistic modes IP header compression for real-time video streaming. IEEE Access, 8, 83155-83166.
10. Feres, C., & Ding, Z. (2019). Low Complexity Header Compression with Lower-Layer Awareness for Wireless Networks. ICC 2019-2019 IEEE International Conference on Communications (ICC),
11. Groza, B., Popa, L., & Murvay, P.-S. (2020). Highly efficient authentication for CAN by identifier reallocation with ordered CMACs. IEEE Transactions on Vehicular Technology, 69(6), 6129-6140.
12. Hermenier, R., & Kissling, C. (2013). Optimization of robust header compression for aeronautical communication. 2013 Integrated Communications, Navigation and

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*

Surveillance Conference (ICNS), Tömösközi, M., Seeling, P., Ekler, P., & Fitzek, F. H. (2017). Performance evaluation and implementation of IP and robust header compression schemes for TCP and UDP traffic in static and dynamic wireless contexts. Computer Science and Information Systems, 14(2), 283-308.
13. Zahra, S. R., & Chishti, M. A. (2020). Packet header compression in the Internet of Things. Procedia Computer Science, 173, 64-69.
14. Jing, S., Zhang, S., & Ding, Z. (2023). Reinforcement learning for robust header compression under model uncertainty. arXiv preprint arXiv:2309.13291.
15. Sandlund, K., Pelletier, G., & Jonsson, L.-E. (2010). The robust header compression (rohc) framework (2070-1721).
16. Cha, H., Shon, T., Kim, K., & Hong, M. (2015). Improving packet header compression with adaptive sliding window size. https://doi.org/10.1109/ICOIN.2015.7057966
17. Chong, Y., & Chee Wan, T. (2011). Header Compression Scheme in Point To Point Link Model Over Hybrid Satellite WIMAX Network (Vol. 2). https://doi.org/10.5121/ijasuc.2011.2308
18. Majanen, M., Koskela, P., & Valta, M. (2015). Constrained application protocol profile for robust header compression framework. Fifth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies, ENERGY 2015.

Corresponding Author: *A. A. Isah*
aaisah@gmail.com
*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria.*