A Multicast Packet Authentication using Signature Amortization Scheme

C.A. Opayinka, B.K. Alese, A. F. Thompson, O. D. Alowolodu, Festus A. Osuolale
Computer Science Department
Federal University of Technology
Akure, Ondo State, Nigeria

Abstract

The major principle of multicast communication is that each data packet sent from one source must reach all the receivers exactly as it originates from the source without any modification on the way. For information to be authentic there is the need to ensure that the information is from where they claim to come from and that security requirements are put in place. In multicast applications, when there are packet losses, the authenticity of the packet will not be verifiable by the receiver, making authentication impossible. In view of this, there is a need to develop an authentication scheme with lower communication cost and one that can be verifiable when packet loss is high. Thus, this research work developed a multicast authentication scheme that makes verification possible when there is packet loss and minimizes the size of the communication overhead. The research simulates a network environment using C sharp programming language and was tested with different messages to a number of receivers. The program was implemented in two phases: the sender's end and the receiver's end. The results show that amortization technique assists in reconstruction of the packet and make verification possible even when there are high losses of packets.

1. Introduction

In this age of growing Information Technology, the internet has been a major means of data transfer. The various method of these data transfer the Unicast (a method of transmitting data from one sender to individual destinations), Broadcast (transmitting the same data to all possible destinations), and Multicast (transmitting data to various interested destinations by using special address assignment).

In today communication, multicast is a more prevalent mode of communication in the networking environment, there are various multicast routing protocols that have been put in place, especially in the Internet Protocol layer [1]. Multicast is a process of data transfer which allows a single copy of packets to be sent by the sender and routed to every receiver within the multicast group with the user of multicast-enabled routers [2].

The major principle of multicast communication is that each data packet sent from one source must reach all the receivers exactly as it originates from the source without any modification on the way. For information to be authentic there is the need to ensure that the information is from where they claim to come from and that security requirements are put in place.

Considering the peculiarity of multicast environment where data packets are sent randomly different cases can occur (collisions, loss of signal or network failure) that can affect data transmission and lead to loss of data there is need therefore, to ensure that data sent from one source to the other actually came from the named source and that authentication is possible even when there is packet loss.

Using digital signatures for authentication, one of the greatest challenges discovered is the computationally intensive nature of the asymmetric-key-based signatures. Authenticating packets one after the other using the sender's digital signature places a significant load on the computation overhead as well as time taken to sign these packets. From previous authentication schemes, some of these problems have been handled one by one: by designing faster signature techniques and amortizing the signature operation over multiple packets but yet the faster these signatures scheme, the higher the communication overhead. There are other issues such as delay on the sender's side and on the receivers side and some of the amortization schemes introduces delay in verification of the sent packet thus when there is a packet loss verification becomes difficult.

Also in multicast applications, when there are packet losses, the authenticity of the packet will not be verifiable by the receiver, making authentication impossible. In view of this, .there is a need to develop an authentication with lower communication cost and one that can still be verifiable when packet loss is high. Thus, this research work intends to develop a multicast authentication scheme that makes verification possible when there is packet loss and minimizes the size of the communication overhead.

2. Literature Review

There are two types of transmission technology mainly used on the internet, Point-to-Point link and Broadcast links. The Point-to-point transmission of packet between one sender and one receiver is known as unicasting while Broadcasting allows packet to be addressed to all destinations by using a special code in the address field, transmission of such packets to a subset of the receivers is called multicasting [3].

Multicasting is a technique that allows copies of a single packet to be passed to a selected subset of all possible destinations. Some hardware, for example Ethernet,

supports multicast by allowing a network interface to belong to one or more multicast groups.

Multicasting can also be defined as a form of communication mechanism for group-oriented applications. It is also a way of sending messages to well-defined groups that are numerically large in size but small compared to the network as a whole. IP multicast saves bandwidth by sending the source traffic on a multicast tree that spans all the members of the group. Therefore, security of multicast communication is an essential requirement for effective utilization of commercial communication applications, it is important to maintain data integrity, secrecy, authentication and access control.

Multicast is a natural way of communicating information and for it to be successfully implemented, it needs various degrees of security considerations, and for example that the information came from the claimed source and that it has not been tampered with on the way [2].

According to O'Mahony, Authentication is the process whereby the receiver of a digital message can be confident of the identity of the sender [4]. It is the process of verifying that information came from a trusted source and that the information have not been tampered with in transit. If a receiver receives a message that comes from sender A, to prove the authenticity of the message the sender applies her secret key to the message before sending it and the receiver applies the sender's public key. Authenticity is guaranteed since the only person capable of producing it is the person in possession of the secret key.

The public-key systems can be used for two purposes: encrypting a message with the recipient's public key to achieve confidentiality, or encrypting a message with the sender's secret key to achieve message authentication. Authentication service is mainly concerned with assuring that a communication is authentic. The function of authentication service is to assure the recipient of the message that it is from the source that it claim to come from. There are two ways to authenticate the origin:

a) Source authentication: This aims to assure that the received multicast messages originate from a source having a specific identity [5]. Source authentication allows a receiver to ensure that the received data is authentic; it originated from the right source and was not modified on the way.

b) Non repudiation allows the recipient to prove the origin of the data to a third party. This is a service which prevents an entity from denying previous commitment or actions

When two parties communicate, there is the need to authenticate the source, Message Authentication Code (MAC) is a symmetric technique used to implement this. It relies on a secret key shared between the two communicating parties. To handle non repudiation of data, a digital signature is used which is based on asymmetric cryptographic technique and this is rather more expensive than Message Authentication Code. Hashing Message Authentication Code (HMAC) is a subset of hashing functions that is used to handle authentication issues. They use a shared secret symmetric key to create the fixed output, called a digital signature. The sent packets cannot be intercepted by eavesdroppers; only the parties that know the key can create and verify the signature for sent data.

2.1. Different methods of Multicast Authentication

There are different methods of multicast authentication that have been proposed by various researchers, they are: the Scalable Reliable Multicast (SRM) and Reliable Adaptive Multicast Protocol (RAMP)

2.1.1. Timed Efficient Stream Loss-tolerant Authentication (TESLA). Timed Efficient Stream Loss-tolerant Authentication (TESLA) is based on Multiple Access Collisions that requires time synchronization between the sender and the receiver [1]. It embeds a MAC in each packet to provide authentication and the corresponding MAC keys are disclosed to the receiver only after a time delay. The delay before the disclosure is chosen long enough so that they cannot be used to forge packets. This scheme is robust against packet loss and scalable, but it requires that the sender and receiver synchronize their clocks within a certain margin. Thus TESLA cannot function without reliable time synchronization.

The main idea of TESLA is to have the sender attach to each packet a MAC computed using a key k known only to it. The receiver buffers the received packet without being able to authenticate it. If the packet is received too late, it is discarded. After a while, the sender discloses k and the receiver is able to authenticate the packet. Consequently, a single MAC per packet allows source authentication, provided that the receiver has synchronized its clock with the sender ahead of time

Asymmetric key cryptography makes use of digital signatures based on asymmetric key cryptography. According to Boneh, one cannot build an efficient (in terms of communication overhead) collision resistant multicast authentication scheme without relying on digital signatures [6]. Many schemes based on asymmetric key cryptography attempt to reduce the computation and communication overhead by amortizing a single signature over multiple packets.

Even if the computational load required for the signature generation is amortized, the communication overhead can be significant if one were to make each packet carry its own authentication information and thus make it individually verifiable. This is the only way to reduce the size of the authentication information and sign multicast packets.

The approach is to divide a stream into blocks and amortize a single signing operation over a block of packets. The authentication tree is computed as follows:

i Packet digests (or hashes) are the leaf nodes.

ii Other nodes of the tree are computed as message digests of their children.

iii The root is the block digest, with the block signature being the signature of the root.

Within the blocks, each packet carries its own authentication information consisting of the signed block digest, the packet position in the block, and the siblings of each node in the path of the packet's corresponding leaf node to the root. To verify a packet, the receiver needs to verify the packet's path to the root and compare the computed block signature with the received one. The advantage is that it improves signing and verification rates but has practical limitations in that it requires a large communication overhead.

But within a block, verification of a packet is dependent on other packets within the block, so communication overhead can be reduced substantially. In this type of an approach, verification of each packet is not guaranteed and instead is assured with a certain probability.

2.1.2. Efficient Multi-chained Stream Signature (EMSS). EMSS uses combination of hash functions and digital signatures to authenticate packets.

The basic idea is that a hash of packet P_1 is appended to packet P_2, whose hash is in turn appended to P_3. If a *signature packet*, containing the hash of the final data packet (i.e., P_3) along with a signature, is sent after P_3 , then non-repudiation is achieved for all three packets. The hash values links the packets so that they form a single string that can be signed by one digital signature but this approach is not robust against packet loss. A single packet loss breaks the chain, and even makes authenticity of the packets preceding the break point impossible.

EMSS handles this by storing the hash of each packet in multiple locations and appends multiple hashes in the signature packet. i.e each packet P_k include hashes $H(P_{k-i})$, $H(P_{k-j})$ of the previous packets P_{k-I}, P_{k-j}

The signature packet, which contains the hashes of the final few packets along with a signature, is sent at the end of the stream to authenticate all the packets. Tolerance to loss can be increased further by sending multiple copies of a signature packet at delayed intervals.

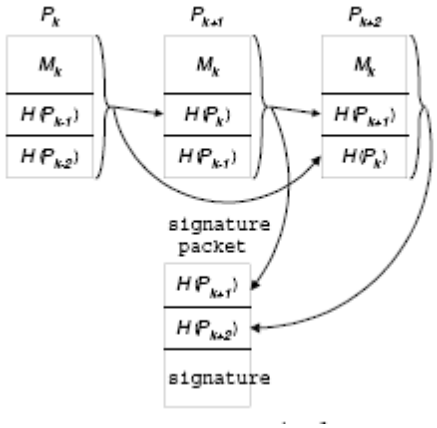

Figure 1. An example of EMSS

To reduce the verification delay at the receiver side, a stream of packets is divided into blocks and all the data packets within the block are chained with multiple hashes followed by an insertion of one or more signature packets.

2.2. Signature Amortisation Scheme

Signature amortization is a means of achieving multicast message authentication by amortising a single signing operation over multiple packet [7].

The common way to authenticate packets during transmission is to sign each individual packet using the sender's digital signature one after the other, at this instance the time taken in authenticating packets one by one and the computation overhead is high. Also the communication overhead required to authenticate the packet one after the other is practically too high to employ, the encoding and decoding process for each hash is takes a lot of time. According to Perrig using universal hash functions or IDA to split the hash value of each packet into multiple pieces before appending them onto other packets certainly produces a more loss-resistant scheme with the same amount of communication overhead [1].

In view of this we consider IDA, an erasure codes with a quadratic decoding time thus problem of sender delay and encoding individual hashes is solved. This is by a way of concatenating all the hashes within the block to form a single piece before encoding is employed. This strategy requires only one encoding and decoding per block which helps to reduce the space overhead.

Thus to authentication multicast streams using signature amortization, the stream is divided into blocks, each block consists of some packets, a single signature is applied for each group of packets, which is known as a block. A sender appends the hash H (Pi) of a packet Pi to specific other packets to achieve robustness against packet loss. For each block the sender then concatenate hashes of specific packets together and signs them. The signed packet is called a signature packet *Psig*. The sender sends a signature packet at the beginning of each block and sent as the first packet to receivers so as to enable them from verifying the received packets.

3. Model Design

In multicast, when a sender transmits the hash of a packet, it is appended to k other packets for increased resistance to loss. For a block consisting n packets, assuming an independent packet loss, the probability that at least one out of the k packets will reach the destination is $1-q^k$, where q is the packet loss probability.

The communication overhead would be kh, where h is the size of the hash. Using the same overhead, one can encode the hash using IDA and append the encoded n segments to the n packets of the block (i.e., each packet in the block would contain one of the encoded segments). The

minimum number of encoded segments needed for reconstruction of the hash is $m = [n / k]$,

The probability that the hash can be reconstructed successfully at the receiver end $Pr_{reconst}$ is given by;

$$1 - \sum_{i=0}^{m-1} \binom{n}{j} (1-q)^i q^{n-i} \quad \text{.....Equation 1}$$

where q is the packet loss probability, n is the number of packets in a block (packet sent), m is minimum number of encoded segments needed for reconstruction of the hash.

3.1. Packet Authentication

To authenticate multicast streams, the stream of packets is divided into blocks with each block consisting of a number of packets. The hash value of each packet is computed, some hash values are appended to other packets, forming what is known as hash chain.

The following steps to authenticate a packet were adopted as follows [2]:

a. If || denote concatenation, dividing a stream of packets into groups (or blocks), a stream G=G1||G2|| .... where each group G_i is a concatenated string of n packets (i.e., $G_i = P_{(i-1)\,n+1} \| ... \| P_{in}$, and each packet $P_i \in \{0,1\}^c$ for some constant c for all groups.

b. A packet hash $H(P_i)$, i =1....,n is computed for each packet using a hash function H.

c. The packet hashes are concatenated to form $F^1 = H(P_1)||...|| H(P_n)$ of size N (i.e., $F1$ consists of N characters). Let bi represent the i-th character in F^1 . (b_i being an eight-bit byte, hence $0 \le b_i \le 255$, $\mathbf{Z}$ 257 or $GF(2^8)$ mode) .

A copy F^1 is stored in a temporary buffer while another copy is divided into blocks of length m as thus: $F^1 = (b_i.....,b_m),(b_{m+1}.....,b_{2m})...,(b_{N-m+1}.....,b_N)$

taking $\mathbf{S}_i = (b_{(i-m)m+1}.....,b_{im})$ $1 \le i \le N/m$.

d. Having a set of n vectors, $\boldsymbol{a_i} = (a_{i1}, ...,a_{im})$, $1 \le i \le N/m$, F_i^1 is processed and divided into n pieces as follows:

$F_i^1 = (a_i .S_1, a_i .S_2 , a_i .S_{N/m})$, $i = 1,....., n$, therefore $|F_1^1| = |F_1|^1/m$.

since $a_i .S_k = a_{i1} .b_{(k-1)m+1} + ...+ a_{in} .b_{km}$

e. The *group hash* $H_G(G_1)$*is* computed by taking the hash of the other copy of F^1: $= H(F^1)$ as follows:

$H_G(G_1) = H(F^1) = H(H(P_1))||...|| H(P_n))$ where $H_G(G_1)$ is the group hash of the first group of packets.

f. The group hash is signed by an asymmetric-key (digital) signature scheme using the sender's private key K_r and denoted as $\sigma(K_r, H_G(G_1))$. This value is again processed by IDA and divided into n segments using the same set of vectors to give

$\sigma_1(K_r, H_G(G_1))... \sigma_n(K_r, H_G(G_1))$.

Note: The signature can be concatenated with F^1 , before applying IDA, so that only one encoding is done per group.

g. Each signature segment (step 7) and hash segment (step 5) are concatenated with the corresponding packet to form an authenticated packet. A group of n authenticated packets combine to form an authenticated group expressed as $\sigma_1(Kr, H_G(G_1)) \| F_1^1 \| P_1 , ..., \sigma_n(Kr, H_G(G_1)) \| F_n^1 \| P_n$

3.2. Packet Verification

At the receiving end, assuming that at least m authenticated packets are received, the receiver can successfully reconstruct F^1 and $\sigma_1(Kr, H_G(G_1))$ from any combination of m packets as follows:

a. Assuming $F_1^1,....., F_m^1$ segments are received, with the m pieces we have:

$$\boldsymbol{A} . \begin{pmatrix} b_1 \\ \\ b_m \end{pmatrix} = \begin{pmatrix} a_1.S_1 \\ \\ a_m.S_1 \end{pmatrix}$$

b. Since **A** is invertible (because of the independence condition on $\boldsymbol{a_i}$,$1 \le i \le n$), $\boldsymbol{S}_1$ can be obtained from

$$\mathbf{S}_1 = \begin{pmatrix} b_1 \\ \\ b_m \end{pmatrix} = A^{-1} . \begin{pmatrix} a_1.S_1 \\ \\ a_m.S_1 \end{pmatrix}$$

c. Computing $\boldsymbol{S}_{2,}.....,\boldsymbol{S}_{N/m}$ by the same procedure in step 2, we can reconstruct F^1 by concatenating these values.

d. Using the same procedure $\sigma(Kr, H_G(G_1))$ is reconstructed

e. All the packets in in G_1 can be verified using F^1 and $\sigma(Kr, H_G(G_1))$.

3.3. Authentication Probability

The authentication probability is an important metric to measure the performance of signature amortization schemes [8]. In multicast setting, packet loss seriously affects the authentication probability.

According to Miner and Alain, Authentication probability is defined as Pr (P_i verifiable/P_i received, where P_i is the i-th packet [9].

For a stream of packet consisting of multiple blocks, authentication probability is defined as:

$$Ps = \frac{1}{S} \sum \frac{number\ of\ verifiable\ packet\ of\ block\ i}{number\ of\ received\ packet\ of\ block\ i}$$

where S is the number of blocks within a stream Verification probability is number of verifiable packet divided by the number of received packets in the stream.

To measure the performance of amortization scheme and the effect of these factors on the authentication scheme, several simulation studies are carried out. The number of hashes appended to a signature packet and number of packets containing the hash of a previous packet have the highest influence on the authentication probability.

4. Results of the New Scheme

In this Scheme, streams of data packet are broken down into blocks and the blocks broken down into data packets. The hashes of all the data packets are signed together with the signature to form the data packet and only one signature packed is encoded for one block of data.

The individual data packet and the hashes are signed with the sender's key and transmitted immediately, so there is no sender's delay. There is only one signing operation over a block of packet thus there is a high resistance against packet loss. The receiver decrypt the packets as soon as they arrive with the public key thus there is no receiver's delay and the verification probability is very high. The authentication probability is also high.

Since we do not have to sign the packet one by one and only one signature packet is required per block of packet the computation and communication overhead is very low.

The following are the samples of different message transmissions and the results recorded. Table 1 shows the result of transmission of multicast packet calculating the loss probability where the number of packet sent is 8.

Table 1. A Table of data transmission to multiple receivers where the number of receivers N=10 and the number of packet transmitted n =8

Receivers	n_{sent}	$n_{received}$	n_f	P_f	Q	P_s=1-q
R1	8	2	6	6/8	0.75	0.25
R2	8	5	3	3/8	0.375	0.625
R3	8	7	1	1/8	0.125	0.875
R4	8	8	0	0/8	0.0	1
R5	8	6	2	2/8	0.25	0.75
R6	8	4	4	4/8	0.5	0.5
R7	8	6	2	2/8	0.25	0.75
R8	8	7	1	1/8	0.125	0.875
R9	8	6	2	2/8	0.25	0.75
R10	8	5	3	3/8	0.375	0.626

Definition of parameters

N is the number of receivers

n is the number of data packets transmitted

n_{sent} is the number of data packets transmitted

n_{recei} is the number of data packets received

n_f is the number of data packets lost

q is the loss probability = n_{sent}-n_{recei} / n

1-q is the probability that at least one out the k packets will reach destination

P_f = probability of failure

P_s == probability of success

Pf + Ps =1

$$\text{Pau} = \text{Pr}\ \frac{Pi\ verifiable}{Pi\ received}$$

The probability that the hash can be reconstructed by the receiver

$1\text{-}n\ (i\text{-}q)^o\ q^{n\text{-}o}\ +\ n\ (1\text{-}q)^1\ q^{n\text{-}1}$

The next data transmission was carried out with the block of data broken into ten packets and the time taken for the data packets to be received is observed. The following table (table 2) shows the results of transmission of packets calculating the loss probability and the arrival time where the number of packet sent is 10.

Table 2. Calculating loss probability and arrival time with 10 receivers (N=10) where the number of packet transmitted n =10

Receivers	n_{sent}	$n_{received}$	n_f	Pf=q	1-q	Arrival time (t_s)	Resend time(t_r)
r1	10	8	2	0.2	0.8	168mls	28mls
r2	10	6	4	0.4	0.6	167mls	11mls
r3	10	0	10	1	0.9	167mls	19mls
r4	10	5	5	0.5	0.5	167mls	15mls
r5	10	9	1	0.1	0.9	167mls	10mls
r6	10	4	6	0.6	0.4	167mls	14mls
r7	10	10	0	0	0	167mls	-
r8	10	1	9	0.9	0.1	167mls	16mls
r9	10	10	0	0	1	167mls	-
r10	10	9	1	0.1	0.9	167mls	9mls

From the Table 2, the report on the arrival time shows there is no delay on the sender's side in transmitting those packets. The graph below illustrates that out of the transmitted ten (10) packets; a good number of packets were received and verified despite the packet loss.

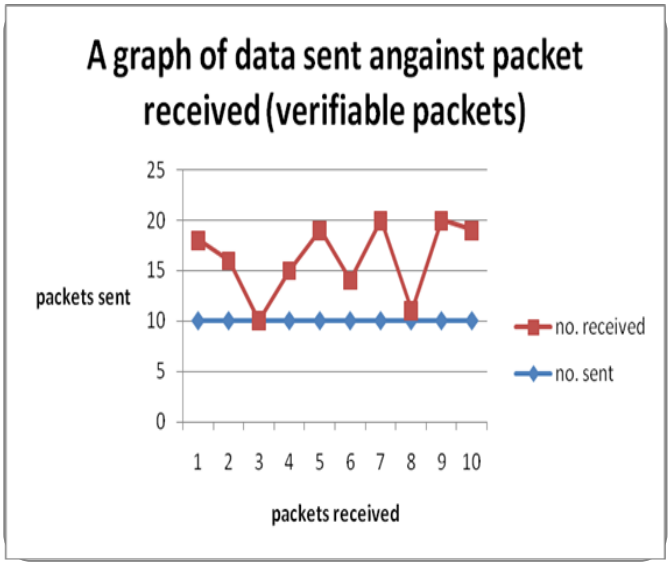

Figure 2. A graph illustrating packets transmitted and the verifiable packets

The next table (table 3) shows the result of transmission of multicast packet calculating the loss probability where the number of packet sent is 19.

Table 3. Calculating loss probability and arrival time with 10 receivers (N=10) and 19 packets (n=19)

Receivers	n_{sent}	$n_{received}$	n_f	Pf=q	1-q	Arrival time (t_s)	Resend time(t_r)
R1	19	17	2	0.10526316	0.89473684	205mls	27mls
R2	19	6	13	0.68421053	0.31578947	205mls	22mls
R3	19	0	19	1	0	205mls	29mls
R4	19	11	8	0.42105263	0.57894737	205mls	25mls
R5	19	9	10	0.52631579	0.47368421	205mls	35mls
R6	19	4	15	0.78947368	0.21052632	205mls	42mls
R7	19	10	9	0.47368421	0.52631579	205mls	24mls
R8	19	8	11	0.57894737	0.42105263	205mls	36mls
R9	19	15	4	0.21052632	0.78947368	205mls	27mls
R10	19	9	10	0.52631579	0.47368421	205mls	20mls

The result from Table 3 shows that even when the loss probability is very high, verification is still highly feasible.

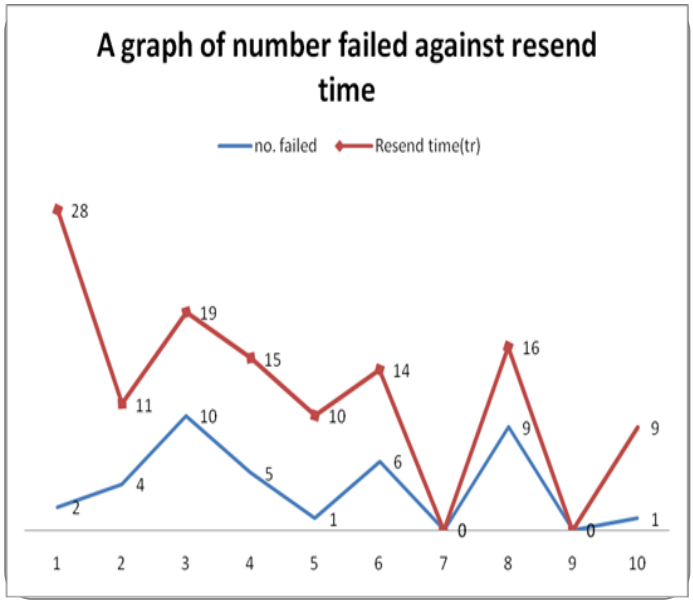

Figure 3. A graph of number of packets failed against resend time

Figure 3 illustrates data in Table 3 where 19 packets were transmitted to 10 receivers, it can be deduced that despite the number of packets that failed, verification is still feasible and on request for retransmission the other packets that are lost are still received.

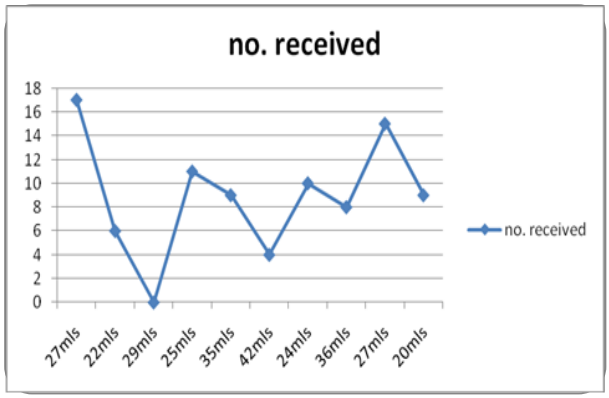

Figure 4. A graph of resend time against number received

From the figures 4 and 5 illustrates data in Table 4.3 where 19 packets were transmitted to 10 receivers.

The next table below shows the result of transmission of multicast packet calculating the loss probability where the number of packet sent is 44 with 10 receivers.

Table 4. Calculating loss probability and arrival time with 10 receivers (N=10) and 44packet packets (n=44)

Receivers	n_{sent}	$n_{received}$	n_f	Pf	1- q	Arrival time (t_s)	Resend time(t_r)
R1	44	31	13	0.29	0.71	654mls	47mls
R2	44	37	7	0.16	0.84	654mls	40mls
R3	44	43	1	0.02	0.98	654mls	54mls
R4	44	24	20	0.83	0.17	654mls	37mls
R5	44	15	29	0.65	0.35	654mls	59mls
R6	44	5	39	0.88	0.12	654mls	49mls
R7	44	13	31	0.70	0.30	654mls	45mls
R8	44	7	37	0.84	0.16	654mls	55mls
R9	44	24	20	0.45	0.55	654mls	57mls
R10	44	32	12	0.38	0.62	654mls	51mls

Table 4 shows that with P_f being the loss probability, with varying degree of loss packets were still verifiable. It also shows that the number of verifiable packet increases as n increases.

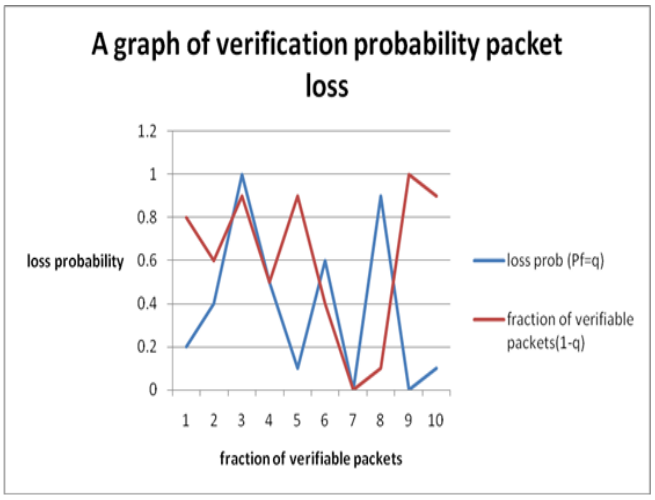

Figure 5. Loss probability against Verification Probability

The next table, Table 5 shows the result of transmission of multicast packet calculating the loss probability with 10 receivers and 30 packets transmitted.

Table 5. Calculating loss probability and arrival time with 10 receivers (N=10) and 30packet

Receivers	n_{sent}	$n_{received}$	n_f	P_f (q)	1-q	Arrival time	Resend time
R1	30	25	5	0.16	0.84	74mls	47mls
R2	30	22	8	0.26	0.74	74mls	47mls
R3	30	10	20	0.66	0.34	74mls	53mls
R4	30	10	20	0.66	0.34	74mls	49nls
R5	30	12	18	0.60	0.40	74mls	54mls
R6	30	8	22	0.73	0.27	74mls	55mls
R7	30	30	0	0	0	74mls	-
R8	30	9	21	0.70	0.30	74mls	52mls
R9	30	25	5	0.16	0.84	74mls	46mls
R10	30	16	14	0.46	0.54	74mls	52mls

Figure 6 illustrates data on table 5, it shows that the number of verifiable packet increases as loss probability increases.

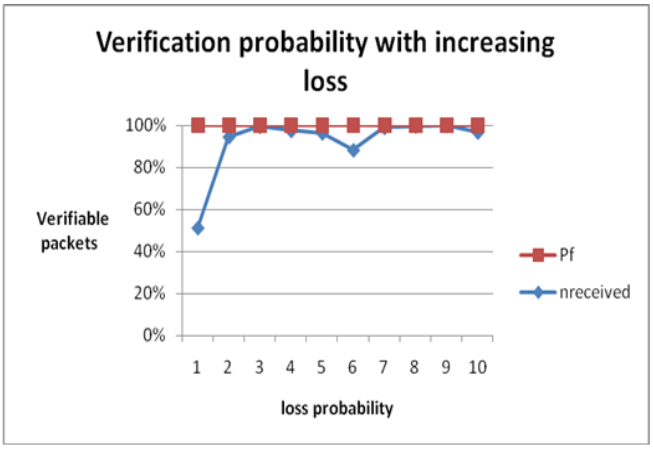

Figure 6. Verification probabilities with increasing loss

The next table, Table 6 shows the result of transmission of multicast packet calculating the loss probability with 10 receivers and the number of packet sent is 30.

Table 6. Calculating loss probability, arrival and resend time with 10 receivers (N=10) and 30 packets (n=30)

Receivers	n_{sent}	$n_{received}$	Nf	Pf	1-q	Arrival time	Resend time
R1	30	1	29	0.96	0.04	231mls	44mls
R2	30	11	19	0.63	0.39	231mls	54mls
R3	30	26	4	0.13	0.87	231mls	39mls
R4	30	17	13	0.43	0.57	231mls	52mls
R5	30	14	16	0.53	0.47	231mls	50mls
R6	30	6	24	0.80	0.20	231mls	60mls
R7	30	23	7	0.23	0.77	231mls	44mls
R8	30	27	3	0.10	0.90	231mls	44mls
R9	30	29	1	0.03	0.97	231mls	40mls
R10	30	15	15	0.5	0.5	231mls	55mls

Average number of packet received = 165/10 =16.5 packet out of 30 packets transmitted.

The from table above out of 30 packets transmitted, the average number of packet received = $=\sum n_{received}$ / no of receivers = (25+22+10+10+12+8+30+9+25+16)/10 = 165/10 =16.5packet

The average packet loss = $\sum n_f$ / no of receivers = (5+8+20+20+18+22+0+21+5+14) /10

= 143/10 = 14.3 packets

Calculating the average loss probability = 14.3/30 = 0.4766

By definition, the packet loss probability is given as q and the communication overhead is given by *kh*, where *h* is the size of the hash, and k is the number of packets that reach the destination.

The signature packet size = 128 bytes

Size of hash = 16 bytes

Number of packets received = 16.5, for n = 30

Communication overhead = 16 x16.5 = 264.8 bytes

Verification probability is the number of verifiable packet divided by the number of received packet, all the packet received are verifiable.

Verification probability and communication overhead can be controlled by changing the number of encoded pieces *n* and the minimum number of encoded pieces *m* needed for decoding. The verification ratio is a good indicator of verification. Verification probability is the probability for a packet to be verifiable given that it is received, for amortization scheme all received packets are verifiable. When it is not amortised, all received packets may not be verifiable because the signature packet for some of these packets may have not arrived at the point of verification.

Size of authentication information increases with number of copies sent per block, Tables 5.2, 5.3, 5.4 and 5.5 shows cases of variation in number of packets sent.

The increase in space overhead (determined by *n* /*m*) only applies to the authentication information, which are the hashes and signatures and not the data itself, with the signature packet *s* = 128 bytes and the hash size *h* = 64 bytes.

The space overhead is only affected by the authentication information which are the hashes and signatures thus as the number of packet in the block increases, it does not necessarily affect the space overhead.

The computation overhead is the number of additional information such as hashes and digital signatures that the sender computes so as to authenticate the packets *s* = 128 bytes and *h* = 64 bytes.

Authentication Probability is given as

$$\text{Pau} = \Pr \frac{Pi\ verifiable}{Pi\ received}$$

where p_i is the ith packet

$$Ps = \frac{1}{S}\sum \frac{number\ of\ verifiable\ packet\ of\ block\ i}{number\ of\ received\ packet\ of\ block\ i}$$

I. Comparism with Previous Schemes

The result of this project work is compared with three other previous research works: the Multiple Connected Chains (MCF), EMSS and Augmented chain.

Multiple Connected Chains (MCF) sends a stream of N message to a number of receivers. The stream of messages is divided into blocks of n packet, the sender appends hash $H(P_i)$ of packet P_i to other packets to handle packet loss. The hashes of all the data packets are also concatenated to form the signature packet P_{sig}. The signature packet is sent first (this allows for immediate verification) and the message M_i is sent with additional authentication information. The packet P_i contains the hashes H (P_i) of and of other v packets such as P_{i+1} and P_{i+jc} where j=1, 2 …….v-1, thus for each block, a number of hashes were concatenated together using the sender's digital key. P_i contains the hashes of succeeding packet, the last packet contains no additional hash and the kth packet contains only a single hash.

This scheme handles packet loss, there is immediate verification, one signature packet per block and the first packet is signed. Since the 1st packet is signed, there is packet delay: the sender buffers the hashes of each packet before sending the signature packet.

In EMSS, according to Perrig, a hash of packet P_i is appended to packet P_2 containing the hash of final data packet P_3 [1]. The signature packet containing the hash P_3 and the signature packet is sent after P_3 achieving non repudiation. The disadvantage of this scheme is that once a packet is lost, the chain is broken and verification becomes impossible thus that hash of each packet are stored in multiple locations and multiple hashes are appended in the signature packet.

In this scheme, the signature packet is sent at the end of the stream, multiple copies of the signature packets are sent at intervals to handle packet loss, there is no sender delay hence no buffering of the packets before sending but there is receiver's verification delay. In attempt to reduce receivers verification delay, the communication overhead is increased.

In Augmented Chain according to Golle, the hashes of the packets are inserted in strategic locations (not randomly as in EMSS) so that the chain of packets formed by the hashes can be resistant to loss [10]. Augmented chain, the stream is also divided into blocks, but each block is constructed with an augmented chain. The chain is constructed in two chains:

i) The chain is formed with the hashes of Packet P_i appended to two other packets.

ii) the rest of the packets p-1 are inserted between each pair of consecutive packet constructed in the first chain.

In this scheme, only the last packet is signed- the hashes appended to the previous packet (left edge), thus there is sender delay because data packets cannot be sent before the signature packet.

The results of this new scheme was compared with previous scheme and it was discovered that this scheme is more efficient in terms of low communication overhead, computational load, resistance to packet loss, sender and receiver's delay and verification probability.

Table 7. Comparative Analysis of Amortisation Schemes

METRICS	**Multiple Connected Chain**	**EMSS**	**Augumented Chain**	**My New Scheme**
Computational Load	One signature packet per block	One signature packet per block	One signature packet per block	One signature packet per block
Communication Overhead	The communication overhead is high	The communication overhead is high	The communication overhead is high	The communication overhead is low
Sender's delay	There is sender's delay	No sender's delay	There is sender's delay	No sender's delay
Receiver's delay	No receiver's verification delay	It has receiver's verification delay	It has verification delay	No receiver's verification delay
Verification probability	Dependent on communication overhead	Dependent on communication overhead-high verification probability	Dependent on communication overhead - high verification probability	Very high probability -dependent on communication overhead
Packet loss	Not robust against packet loss	Not robust against packet loss	robust against packet loss	Very robust against packet loss

5. Conclusion

This work provide a model for authenticating packets in a multicast environment where packet move randomly, relying authentication schemes that signs packet one after the other the other is no match for time and cost management. Considering the inevitability of packet loss in a network environment and that the authenticity of the packet will not be verifiable by the receiver when there is high packet loss, an attempt was made to develop an authentication with lower communication cost and one that can still be verifiable when packet loss is high

The research simulates a network environment using C sharp programming language was tested with different messages to a number of receivers. The program was implemented in two phases: the sender's end and the receiver's end. The results show that amortization technique assists in reconstruction of the packet and make verification possible even when there are high losses of packets. When compared with other authentication schemes, the probability of verification is much higher and without introducing any delay in verification.

6. References

[1] Perrig A, Canetti R, Tygar J. D., and Song D. (2000) "Efficient authentication and signing of multicast streams over lossy channels," IEEE Symposium on Security and Privacy, pp.56–73.

[2] Steiner, J. G., B. Clifford Neuman, and J. I. Schiller, .Kerberos: An Authentication Service for Open Network Systems,. Proc. Usenix Conference, Dallas, TX, February 1988, pp. 191.202, http://nii.isi.edu/info/kerberos/ documentation.html.

[3] Andrew S. Tanenbaum, Distributed Systems: Principles and Paradigms © 2003 Pearson Education, Inc. Publishing as Prentice Hall PTR Upper Saddle River, New Jersey 07458

[4] Donal O'Mahony, Michael Peirce, Hitesh Tewari, Electronic Payment Systems for Commerce Second Edition, 2001, ISBN 1-580532683

[5] Perrin. Chad (December 5, 2007). "Use MD5 hashes to verify software downloads". TechRepublic. Retrieved March 2, 2013.

[6] Boneh, D. Durfee, G. and Franklin, M. (2001) "Lower bounds for multicast message authentication," Eurocrypt 2001, pp. 437–452.

[7] Douglas E. Comer, (2000) "Internetworking with TCP/IP" Vol 1. , Principles, Protocols and Architecture, Prentice Hall Inc. , 4th Edition IEEE (2002) Symposium on Security and Privacy (S&P.02) 1081-6011/02

[8] Abuein Quasai, Shibusawa Susumu (2005) "Authentication Probability of Multiple Connected Chains Model for Signature Amortization", IEIC Technical Report (Institute of Electronics, Information and Communication Engineers) ISSN:0913-5685, VOL.105;NO.79(IA2005 1-6);PAGE.13-18

[9] Alain Pannetrat, Refik Molva, Efficient Multicast Packet Authentication, Institut Eurécom {Alain.Pannetrat@eurecom.fr, Refik.Molva@eurecom.fr}, 2005 IEEE International Symposium on Signal Processing and Information Technology, 0-7803-9314-7/05/2005 IEEE 918

[10] P. Golle and N. Modadugo. Streamed authentication in the presence of random packet loss. In NDSS 2001., 2001.

