ESAC Based Channel Aware Routing Using Route Handoff

Dr. V.kavitha,M.E,Ph.D.

Asst.Professor, Dept of CSE,Anna University of Technology, Tirunelveli

S.Balaji,

PG Scholar,Dept of CSE, Anna University of Technology, Tirunelveli

Abstract

An ad hoc network is a collection of mobile stations forming a temporary network without the aid of any centralized coordinator. Internal threats due to changes in the node behavior that target the routing discovery or maintenance phase of the routing protocol (CA-AOMDV) which applies a preemptive handoff strategy to maintain reliable connections by exploiting channel state information. Using the same information, paths can be reused when they become available again, rather than being discarded. And security challenges can however lead to insecure communication in MANETS. We proposed a model that found the improper behavior of the nodes and eliminated them. Also it provided secure routing mechanism of sharing messages between source and destination by adding information about sender and preserves the message content by computing hash function which prevents not only known attacks but also maintains the integrity of data. The reason we chose ESAC was because it typically enabled very high coding efficiency and provided better security.

Introduction

Arithmetic coding is a popular and efficient lossless compression technique that maps a sequence of source symbols to an interval of numbers between 0 and 1[1]. The encoder produces a code stream of bits that uniquely represents the interval; the decoder then maps the code stream to the original source sequence. In arithmetic coding, an entire source sequence is mapped to a single code stream. Therefore, a single error in an arithmetic code stream often causes error avalanches at the decoder, rendering the decoded code stream useless. *Full resynchronization* occurs when the decoder can exactly determine the initial *b* bits of the code stream. In this case the entire original source sequence can be reproduced exactly. *Partial resynchronization* occurs when the decoder only determines the current interval after b bits of the code stream.

Binary arithmetic coding involves recursive partitioning the range (0, 1) in accordance with the relative probabilities of occurrence of the two input symbols [2]. The overall length within the range (0, 1) allocated to each symbol is preserved, but the traditional assumption that a single contiguous interval is used for each symbol is removed. A key known to both the encoder and decoder is used to describe where the intervals are "split" prior to encoding each new symbol. a key-based interval splitting arithmetic coder can be implemented using techniques similar to those used in traditional arithmetic coding and can benefit from the same optimizations for speed, finite precision, etc. The main difference lies in the doubling of the number of intervals, which doubles the memory requirement because the upper and lower limits of two intervals must be maintained. The number of potential split locations—and therefore the level of secrecy— are determined in part by the precision of the key. The key can be used to directly identify split locations, or it can reference locations in a table known to both the encoder and decoder. The splitting produces encryption, the level of which is a function of the specific attributes of the key and the encoded sequence.

A chaos-based adaptive arithmetic coding-encryption technique has been designed, developed and tested and its implementation has been discussed[3]. For typical text files, the proposed encoder gives compression between 67.5% and 70.5%, the zeroth-order compression suffering by about 6% due to encryption, and is not susceptible to previously carried out attacks on arithmetic coding algorithms. Characteristics of chaotic systems like ergodicity, mixing and sensitivity to initial conditions have been seen as analogous to and/or giving rise to confusion and diffusion balance and avalanche property , known properties of a good cipher. The proposed algorithm is resistant to chosen plaintext attacks because of the following. The model dynamically reorders the frequency of the input symbols according to the coupled chaotic system, and depends on all text that has been coded since the initialization of the model. The model dynamically reorders the frequency of the input symbols according to the coupled chaotic system, and depends on all text that has been coded since the initialization of the model. The output from the engine is in the form of variable sized words and the individual bit output corresponding to inserted symbols cannot be determined.

For SAC,a series of permutations are applied at the input and the output of the encoder[4]. The overall system provides simultaneous encryption and compression, with negligible coding efficiency penalty relative to a traditional arithmetic coder. The system consists of a first permutation step applied to the input sequence, arithmetic coding using interval splitting, and a second permutation step applied to the bits produced by the coder. A key sequence is input to a key scheduler which in turn provides key information to both permutation steps and to the interval splitting arithmetic coder. The key scheduler utilizes information from the split AC encoder output. The system offers both compression and security, and thwarts all known attacks aimed at obtaining information about the input or output permutation or the interval splitting keys.

Compared with the original SAC,[5] improved SAC remove the input symbol permutation step. In addition, improved SAC replace the output codeword permutation step with a simple bit-wise XOR step. The design of the key scheduler is very flexible; we can either use the keyed XOR operation as in or other highly efficient pseudorandom number generators. The only private information of the improved system is the seed used in the key scheduler, which is assumed to be of length 128 bits, so as to ensure high enough level of security. It resists the adaptive chosen-cipher text attack and can be conveniently incorporated with the context- basedcoding.

In the present paper, we have identified the major flaw in the previous algorithm that is it deals with the security of data before transmission. That is when the attacker saw the codeword he shouldn’t get nothing from the codeword. Any cryptography concept must satisfy its own four goals that are confidentiality, authentication, Data integrity and non-repudiation. So we going to make the SAC to satisfy these goals to escort the data from known attacks and the attacks happened in the transmission like Man-in-Middle attack, Replay attack etc.But while in the transmission it didn’t talk or assured about detecting the data alteration or data loss and points the intruder who involved in the transmission unauthorized. We have framed our architecture by adding our system before and after the SAC process which transforms the text into numbers and the preservation and data integrity is maintained by adding hash function and sender’s identity during and after the transmission.

Review of SAC

In the SAC two permutation steps are applied to the input symbol sequence and the output codeword, respectively. Let S=s1,s2,…sN be the symbol sequence to be encoded. The encoding procedure of the SAC is shown as follows[5].

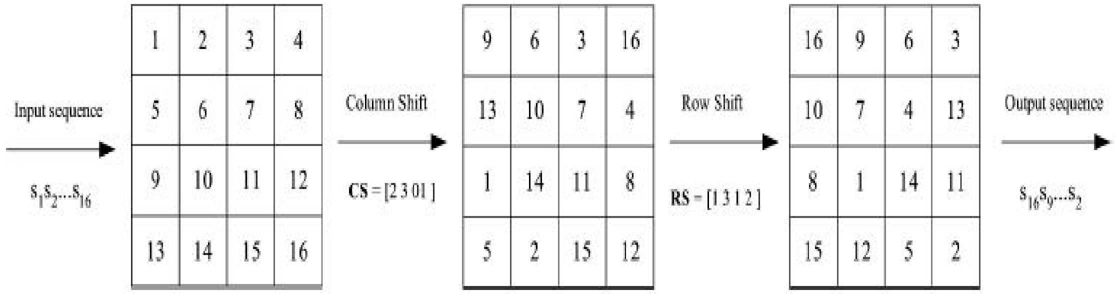
Step 1) Map the sequence S into a block having four columns and N/4 rows.

Step 2) Perform two key-driven cyclical shift steps to the resulting symbol block, and read out the data in raster order to obtain the permuted symbol sequence S1.

In Fig. below, we show an example of the cyclical shift steps, where S is of length 16 and the key vectors controlling the column CS=[2 3 0 1] and the row shift offsets RS=[1 3 1 2] are and, respectively.

Step 3) Input to the ISAC encoder and obtain the intermediate codeword C=c1,c2, cNc.

Step 4) Set C=c1,c2,……….cNc-4 by removing the last four bits of . Map into a block having four columns and [(Nc-4)/4] rows.



Step 5) Perform the first round of shifts to the resulting bit block, which consists of two key-driven cyclical shift steps, one operating on columns and the other on rows. Here, the key vectors controlling the shift offsets depend on the last four bits of C.

Step6)Reappend to the resulting bit block.

Step 7) Perform the second round of shifts to the resulting bit block, which consists of two key-driven cyclicalshift steps, one operating on columns and the other on rows. Here, the key vectors controlling the shift offsets are fixed for all.

Step 8) Read out the data in raster-order from the resulting block to obtain the final bit stream.

Improved SAC

The improved system which mainly consists of two parts:

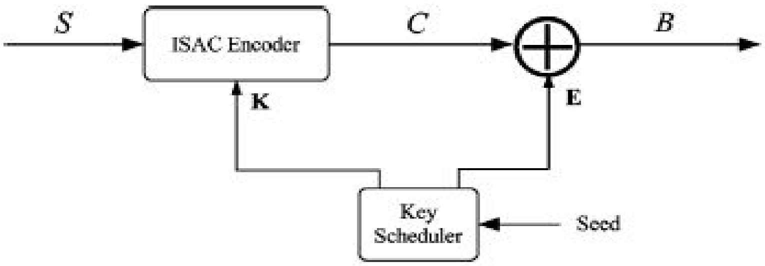
1) an ISAC encoder

2) a key scheduler.

Let the symbol sequence S=s1,s2,…sN be that is to be encoded. The basic steps of performing
the encoding are as follows.

Step 1) Encode S using an ISAC encoder with splitting key K vector . Denote the generated bit stream C=c1,c2,……….cNc as , where Nc satisfies (1).

Step 2) Perform bit-wise XOR operation between C and a key stream E, where has the same length as C. In other words, the final bit stream $B = \tilde{C} \oplus \mathbf{E}$.



Compared with the original SAC, we remove the input symbol permutation step. In addition, we replace the output codeword permutation step with a simple bit-wise XOR step. The only private information of the improved system is the seed used in the key scheduler, which is assumed to be of length 128 bits, so as to ensure high enough level of security.

Review of AODV& AOMDV

AODV

AODV (Adhoc On-Demand Distance Vector routing protocol) is a single-path, on-demand routing protocol When a source node, ns, generates a packet for a particular
Destination node, nd, it broadcasts a route request (RREQ) packet. The RREQ contains the following fields:

<source IP address,
source sequence number,
broadcast ID,
destination IP address,
destination sequence number,
hop-count>,

An intermediate node only processes a RREQ if it has not received a previous copy of it. If an intermediate node has a route to nd with destination sequence number at least that in the RREQ, it returns a route reply (RREP) packet, updated with the information that it has. If not, it records the following information: source IP address, source sequence number, broadcast ID, destination IP address and expiration time for reverse path route entry, and forwards the RREQ to its neighbors.

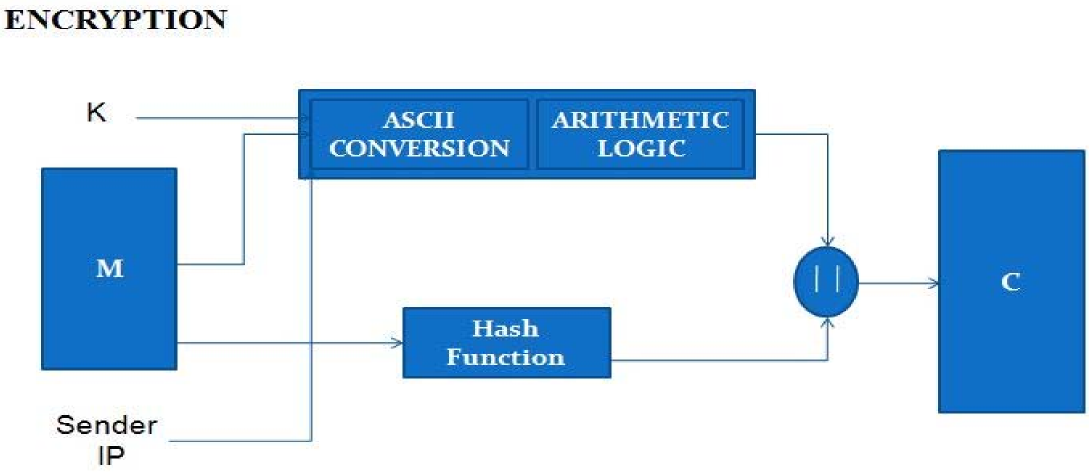
AOMDV

The key distinguishing feature of AOMDV over AODV is that it provides multiple paths to nd. AOMDV uses the notion of advertized hop-count to maintain multiple paths with the same destination sequence number. the routing table entry is slightly modified to allow for maintenance of multiple entries and multiple loop-free paths. First, advertized hop-count replaces hop-count and advertized hop-count is the maximum over all paths from the current node to nd, so only one value is advertized from that node for a given destination sequence number. Second, next-hop IP address is replaced by a list of all next-hop nodes and corresponding hop-counts of the saved paths to nd from that node, as follows:

<destination IP address,
destination sequence number,
advertized hop-count,
route list: {(next hop IP 1, hop-count 1),
(next hop IP 2, hop-count 2), . . . },
entry expiration time>.

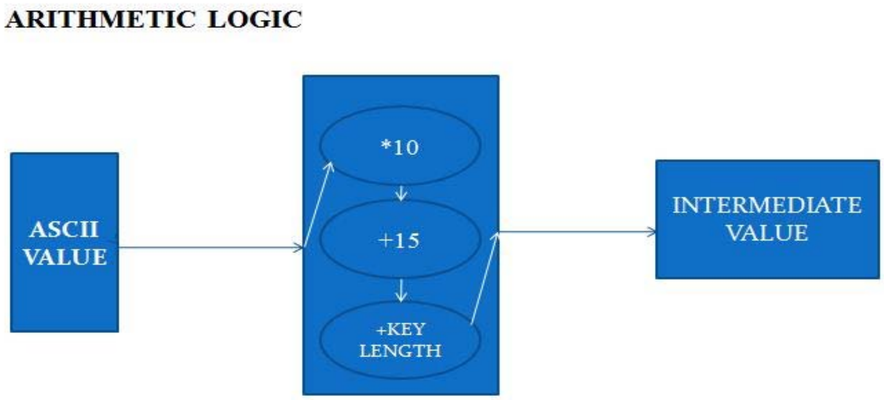
Security Considerations: ESAC

System Framework



Encryption

1. Create the message to be sent and give the key to encrypt the message.
2. Retrieve the sender’s IP and calculates the hash code for the message.
3. Perform ASCII conversions for these three values and Fed ino Arithmetic logic.
4. In the Arithmetic logic, the ASCII code undergoes some arithmetic calculations and at the end key length is added.



5. Append a random value at the end of each parameter. For message the value ranges from 0 to5.For key the value ranges from 6 to 10.For Sender ip the value ranges from 11 to 15.
6. After creating this codeword, append the hash value at the end of the cipher and transmits the message.

Centralization

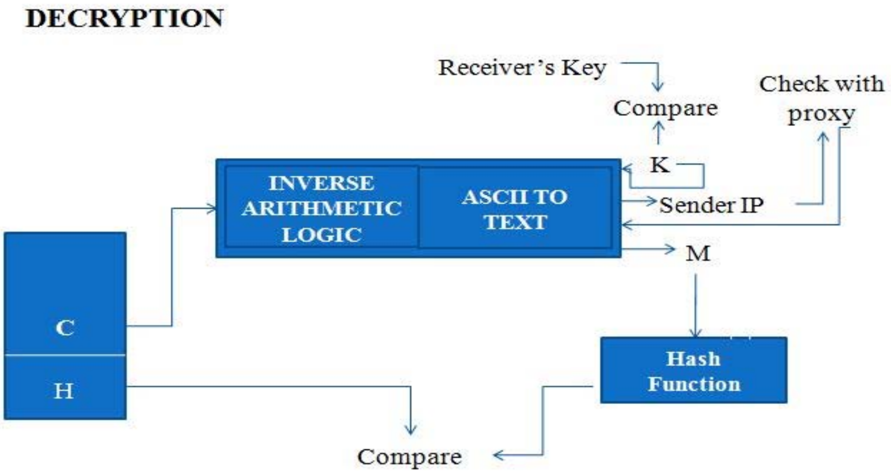
1.The transmitted messages are received by the proxy and retransmitted to the intended client after checking its IP with its list.

2.It maintains a database which has the legitimate users IP along with their username and password.

3.Every time it receives the message, it retrieves the sender IP from the cipher and try to find the match with the list of IP stored in the database.

Decryption

1. After receiving the cipher from Proxy, retrieve the hash value from the cipher and store it in separate area.
2. Fed the remaining cipher into inverse arithmetic logic and perform text to ASCII conversion.
3. Separate the remaining three parameters and store in different area for easy retrieval.
4. Get the receiver key to decrypt and try for a match with the key retrieve from the cipher.
5. If it matches, then retrieve the sender IP and Check with the proxy. If it matches go for next checking task.
6. Then calculate the hash code for the decrypted message and try for a match with the hash code retrieve from the cipher.
7. If all the above conditions are satisfied, then displays the message to the receiver.



Message Creation

Make sure that server is running by getting the response from it to the client who wishes to start the transmission. After getting the response create the message to be transmitted by using ASCII key conversion and Arithmetic Logic. Then get the key known to both the sender and the receiver to encrypt the created message. Calculate the Hash code of the Original message and append to the above created cipher. Then encrypt the key as in the method message encrypted. Append the encrypted key to the above created cipher. Get the source ID of the client who is transmitting and destination of the client who is going to receive. The parts in the cipher like message, key, sender ID, destination ID and hash value of the original message are separated by random numbers in different ranges. After finishing all this process, transmit the message.

Monitoring Transmission

Centralised proxy is in ready state to receive the messages and to transmit the message. Once it receives the message, it strips the source ID and destination ID.Then it checks the source ID with the database stored in proxy for getting the match. If it matches precede the transmission and forward the message to the intended client. If it doesn't match with the ID stored in the database, deny the transmission of the client who just

]

transmitted. And ask the client if he has an interest in registering this group. If he accepted, then ask the user to enter his details like username, password etc. After entering his username check its availability with the database stored in the centralised proxy. And then stipulates the user to choose the password between 8 to 16 characters. Once the details verified, Send the Acknowledgement for the accepted registration to the intended client.

Key creation and transmission

If the client wishes to transmit, he has to send the ready message to the centralised proxy. Proxy in turn generates a random key based on the client password who wishes to start the transmission. Send this random key to the client who is ready to transmitting. Then the sender in turn sends the key to the intended client. After receiving this message to the proxy, the sender key is encrypted with the intended client's password and forward the encrypted the key to the destination. Once the receiver receive this encrypted the key, using his password he can decrypt the key transmitted by the sender. Then the communication starts. On further onwards all the messages are encrypted and decrypted with the temporary key known by both sender and receiver till the session has finished. Once the session is terminated the temporary key will be expired. For the next session another key will be generated for the same client.

Message Reception

Once the client receives the message from the proxy it decrypts the message with the receiver's key. After it decrypts it separates the parts of the cipher like original message, key, senders ID and the hash value of the original message. Compare the decrypted key with the entered receiver key. If it matches checks the sender ID with the database stored in the centralised proxy. If the result is Positive then calculate the hash value of the decrypted message and compares with the decrypted hash value of original message. If it matches displays the message to the receiver.

Algorithm

```
Input: M-Message, K-Key
Function encrypt () return value
Append random no (<5)
B[]=Alogic(IP,k.length)
Append random no (<10 & >5)
C[]=Alogic(k,k.length)
Append random no (<15 & >10)
Append hash (m)
Merge the arrays
Return merged array
Alogic(S,sk.length)
Begin
For i=0 to s.length do
A[]=ASCII(s)
```

```
A[]=10a[]+15+sk.length
end
```

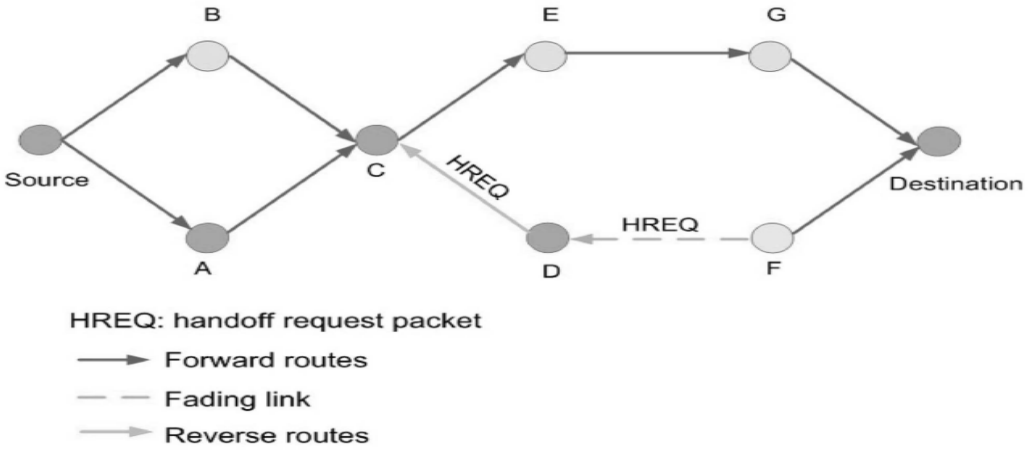
Description

1. Get Data to be sent.
2. Perform ASCII Conversion and Alogic to the above data.
3. Append Random no to the above result in the range between 0 to 5.
4. Retrieve Sender IP and Key from sender.
5. After the conversion, Random no is appended to the key and Sender IP in the range between 6 to 10 and in the range between 11 to 15.
6. Calculate the Hash value of the original message and Append the value at the end of the cipher.

Optimization: CAR

The only demerit of AOMDV is considering only no of hops for a path to reach the destination and ignores path stability. Selected paths tend to have a small number of long hops meaning that nodes are already close to the maximum possible communication distance apart, potentially resulting in frequent link disconnections. Further, channel conditions are idealized with the path-loss/transmission range model, ignoring fading characteristics inherent in all practical wireless communication environments.

ECAR(Esoteric Channel Aware Routing) address this deficiency in two ways. In the route discovery phase, we utilize the ANFD of each link as a measure of its stability. In the route maintenance phase, instead of waiting for the active path to fail, we preempt a failure by using channel prediction on path links, allowing a handover to one of the remaining selected paths. This results in saved packets and consequently smaller delays.



Route Identification

It incorporates channel properties for choosing more reliable paths. ECAR uses the ANFD as a measure of link lifetime. The duration, D, of a path is defined as the minimum ANFD over all of its links,

$$\mathcal{D}_{\min}^{i,d} \triangleq \min_{\zeta \in \text{path\_list}_i^d} \mathcal{D}_{\zeta},$$

where h is link number, and H is number of links/hops in the path. Before forwarding a RREQ to its neighbors, a node inserts its current speed into the RREQ header so that its neighbors can calculate the link ANFD Thus, all information required for calculating the ANFD is available via the RREQ.

Path list^{d}_{i} is the list of all saved paths between nodes ni and nd. The route Identification algorithm in ECAR is a slight modification of that of AOMDV. If a RREQ or RREP for nd at ni, from a neighbor node, nj, has a higher destination sequence number or shorter hop-count than the existing route for nd at ni, the route update criterion in ECAR is the same as that in AOMDV. However, if the RREQ or RREP has a destination sequence number and hop-count equal to the existing route at ni but with a greater $D^{i,d}_{min}$, the list of paths to nd in ni's routing table is updated.

Route Maintenance

Using prediction and handoff to preempt fading on a link on the active path, disconnections can be minimized, reducing transmission latency and packet drop rate.

All nodes maintain a table of past signal strengths, recording for each received packet, previous hop, signal power, and arrival time**.** The handoff process is implemented via a handoff request (HREQ) packet.

Prediction Length

In ECAR,a given node may have multiple paths to the destination, each with a different next hop node. If an intermediate node has multiple paths to the destination, upon receiving an HREQ it can immediately switch from the active path to a good alternative one, without further propagating the HREQ. Therefore, the time needed to implement a handoff in CA-AOMDV is the duration, in terms of the discrete time interval Δt, for the HREQ to be propagated to the fading link uplink node.

A suitable prediction length in corresponds to the number of discrete time intervals, Δt, for transmission of a HREQ between nj and ni, which can be approximated by using the data propagation time T^{i}_{j} from nj to ni, with

$$\psi = \mathrm{round}\left(T^{i}_{j}/\Delta t\right)$$

Where "round" is the integer rounding function.

Handoff Trigger

Route handoff is triggered when a link downstream node predicts a fade and transmits a HREQ to the uplink node. If a fade is predicted at either time, the receiver checks whether the link is at breaking point with respect to distance. The HREQ registers the following fields: source IP address, destination IP address, source sequence number, fade interval index, long term fading indicator, AFD, and v_{T}^{max} .**Handoff Table to Avoid Duplicate HREQs**

Each node maintains a local handoff table. Each entry includes: source IP address, source sequence number, destination IP address, and expiration timeout. expiration timeout indicates when a path is expected to be available again (out of the fade) and is set to the maximum AFD of all currently faded links with paths through that node to a particular ns**.** Whenever a node receives a HREQ targeting a particular ns, it checks its

Handoff table for an entry relating to that ns. The handoff table is updated if no entry exists for that ns, if the new HREQ has a longer AFD or if the existing entry is stale due to the expiration timeout having expired. If any unexpired entry is found for that ns with the same or higher source sequence number, the HREQ is dropped.

Forwarding the HREQ

Any node receiving a nonduplicate HREQ checks for alternative paths to nd. If not,it propagates the HREQ. Otherwise, if it has one or more “good” alternative paths to the nd, it marks the fading path indicated in the HREQ as dormant, setting the handoff dormant time in its routing table entry for that path to the AFD recorded in the HREQ. The HREQ is then dropped. If a fade is predicted on the active path, a nondormant alternative path to nd is then adopted prior to the onset of link failure. The dormant path is retained for use when the fade is over, reducing path discovery overhead.

Conclusion

In the previous system, it is immune against all the known attacks including the adaptive chosen-cipher text attack. In this system, the techniques we have used not only prevent the attacks overcome in the previous systems but also the attacks happen during the transmission also detected and prevented. we are not only securing the data from the transmission attacks like third party attack but also transmitting the data in a optimized way that achieve better efficiency.In the future we are going to develop a better scheme using the same technique to improve the compression efficiency.

References.

[1] Jiantao Zhou,Oscar C. Au,, and Peter Hon-Wah Wong,“Adaptive chosen-ciphertext attack on secure arithmetic coding” *IEEE Trans. Signal Process., vol. 57, no. 5, pp. 1825–1838, May 2009.*

[2] H. Kim, J. T. Wen, and J. D.Villasenor,“Secure arithmetic Coding,” *IEEE Trans. Signal Process., vol. 55, no. 5, pp. 2263–2272, May 2007.*

[3] J. T. Wen, H. Kim, and J. D. Villasenor, “Binary arithmetic coding with key-based interval splitting,” *IEEE Signal Process. Lett., vol. 13,*pp. 69–72, Feb. 2006.

[4] R. Bose and S. Pathak, “A novel compression and encryption scheme using variable model arithmetic coding and couple chaotic system,”*IEEE Trans. Circuits Syst. I, vol. 53, pp. 848–857, Apr. 2006.*

[5] P. W. Moo and X. Wu, “Resynchronization properties of arithmetic coding,” in *Proc. IEEE Int. Conf. Image Process., Oct. 1999, pp.*545–549

Dr.V.Kavitha obtained her B.E degree in Computer Science and Engg in 1996 from MS University and ME degree in Computer Science and Engineering in 2000 from Madurai Kamaraj University. She is the University Rank Holder received PhD degree in computer science and Engg from Anna University Chennai in 2009. Right from 1996 she is in the Department of various designation ns. Presently she is working as Asst. Prof in the Department of CSE at Anna University Tirunelveli..In addition She is the Director In-Charge of University Nagercoil campus Nagercoil. Currently, under her guidance ten Research Scholars are pursuing PhD as full time and part time. Her research interests are Wireless networks Mobile Computing, Network Security, Wireless Sensor Networks, Image Processing, Cloud Computing .She has published man y papers in national and International journal in areas such as Network security, Mobile Computing, wireless network security, and Cloud Computing. She is a life time member of ISTE

S.Balaji,M.E,MISTE is currently working as a lecturer dept of CSE in SCAD College of engineering and Technology,Tirunelveli His research interests are Wireless networks Mobile Computing, Network Security, Wireless Sensor Networks, Cloud Computing .He has presented many papers in National and International conferences in Network security, Mobile Computing, wireless network security, and Cloud Computing. And also his methodology of teaching about TCP & UDP is hosted on Wipro Mission 10x portal. He is a life time member of ISTE