DESIGN AND ANALYSIS OF A SIMPLE LIGHT WEIGHTED SECURE SCHEME FOR MONITORING VARIOUS ATTACKS IN A WSN

KONA DIVYA [#1] , A.V.D.N. MURTHY [#2]

[#1] M.Tech Scholar ,Department of CSE
Pydah College of Engineering and Technology, Gambheeram Village,
Anandapuram, Mandal-531163. Visakhapatnam, AP, India.

[#2] Assistant Professor , Department of CSE
Pydah College of Engineering and Technology, Gambheeram Village,
Anandapuram, Mandal-531163.Visakhapatnam, AP, India.

ABSTRACT

Now a day's wireless sensor networks (WSN) have achieved a lot of user's attention towards its usage and deployment for communication. Although it gained a lot of user's attention in terms of information exchange and efficiency of retrieving data from large distances, still it faces some limitations like data losses and packet losses. The data in a wireless sensor networks is always streamed from a various sources through an intermediate processing nodes that aggregate information. During the data communication a malicious node may try to introduce a set of additional nodes inside a network or compromise existing nodes in a network. Hence to achieve high data trustiness is very crucial for correct decision-making, so we designed a new concept like data provenance, which plays a key factor in evaluating the trustiness of sensor data. In this paper, we for the first time have proposed a simple light weighted scheme to securely transmit provenance for sensor data. In the proposed paper we mainly use the Cuckoo Filter to encode the data provenance and also we introduced the efficient mechanisms for data provenance verification and reconstruction at the base station. Also as an extension for this current paper we also proposed a mechanism to identify the packet drop attacks which was created by malicious data forwarding nodes.

Key Words: Data Communication, Data Aggregation, Trustiness, Cuckoo Filter, Compromised Node, Data Provenance, Packet Dropping Attacks.

I. INTRODUCTION

Now a day's security plays a very vital role in each and every organization like banking, software, shopping malls, E-commerce, Schools, Hospitals and so on. As security plays a very prominent role a lot of users try to access the contents illegally and they want to misuse the

content during transmission. There are several ways of creating attacks either physical attack or non-physical attack. Physical attacks are those which are created based on a hacker and the content will be damaged or modified or lost during the transmission from a selected source node to valid destination. These attacks create physical damage for the data which is been transferred. But non-physical attacks come under a threat model which willn't damage the original content but just creates some delay while transmission. One among the physical attack is forgery attack which will try to do some modification or change in the content of sender and receiver during the data transfer and it will physically change the data content [1]. As this attack may lead a physical change in the content to be send, this attack come under physical mode.

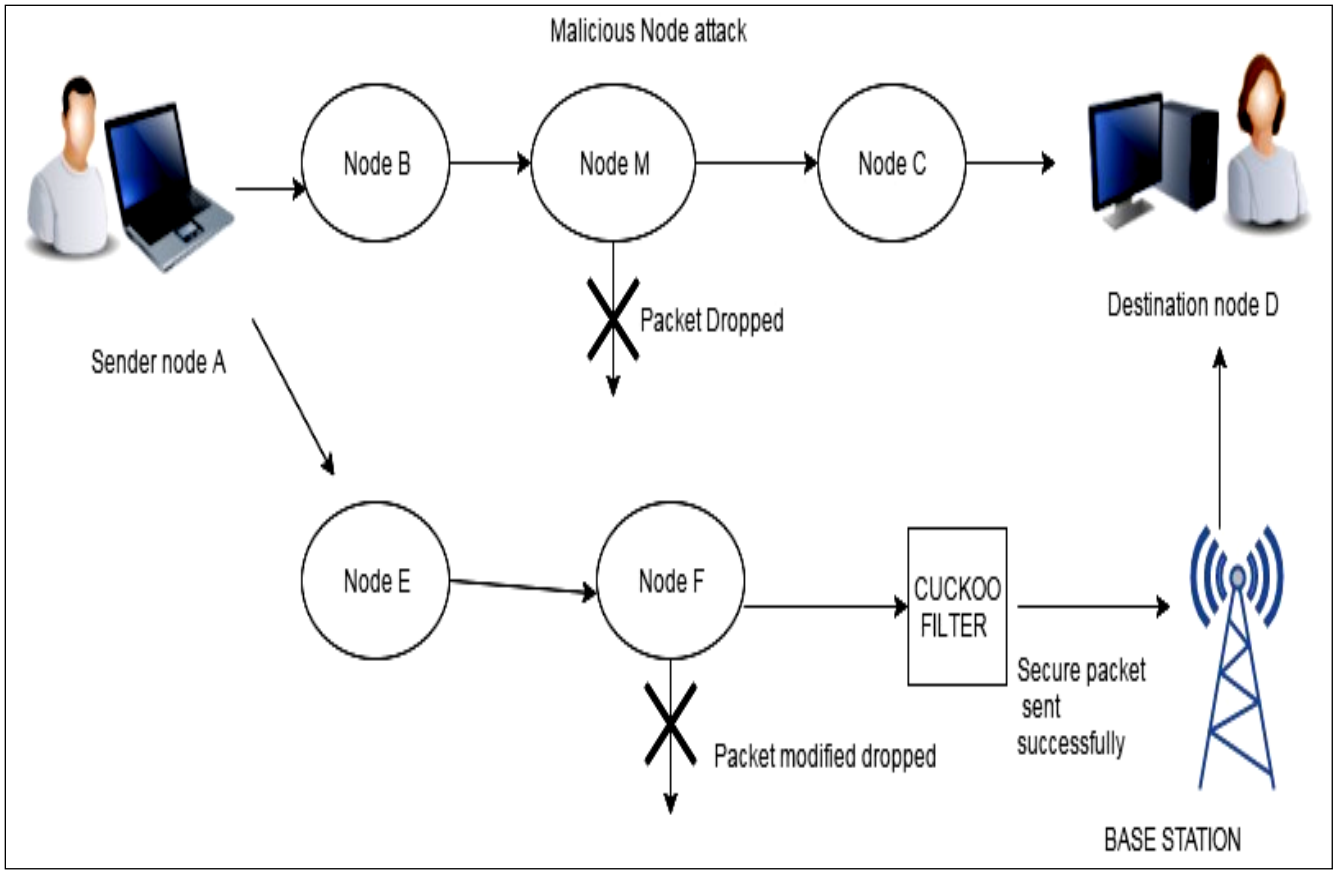

FIGURE 1. REPRESENTS THE ARCHITECTURE OF A VARIOUS ATTACKS IN A WIRELESS SENSOR NETWORKS

From the above figure 1, we can clearly find out the architecture of our proposed application, where the sender node A choose a text file and split the file into a number of packets of equal size. Now we try to start the intermediate nodes that are available between sender and receiver, where all the intermediate nodes receive the incoming packets from the sub-sequent nodes and forward to the next available nodes until it reaches to the destination. Here we apply cuckoo filter in the network during data transmission, so the cuckoo filter will try to avoid the packets not to be passed from the attacked nodes, instead it will be send from the active nodes which are not attacked by any un-authorized users. From the above figure Node M is dropped by an attacker and Node F is attacked by packet modifier during data transmission. If a sender A

sends a message through the networks, the cuckoo filter will try to avoid Node M and Node F nodes that are affected by the intruders or attackers.

In the field of information technology or computing, a forgery attack is an attempt to turn the machine or network resource changed or altered to its intended users, such as to interrupt the data which is being transmitted for the end user and this will create some sort of attack on the data which is transmitted [2],[3]. Along with this forgery there is also a chance of creating an attack like cross site and phishing in which they also comes under physical attack by damaging the content of end users. Generally a lot of criminals or intruders often target valuable sites like bank servers or credit card payment gateways or online shopping gateways and they try to make the process go with some modified values by changing the recipient account details or amount specified limit and then try to send those into their account, which in turn leads to attack [4],[5].Sensor networks are employed in varied application domains, like cyber physical infrastructure systems, environmental watching, power grids, etc. knowledge are made at an over sized variety of detector node sources and processed in-network at intermediate hops on their thanks to a Base Station (BS) that performs decision-making. The range of knowledge sources creates the requirement to assure the trustiness of knowledge; specified solely trustworthy info is taken into account within the call method. Knowledge birthplace is a good technique to assess knowledge trustiness, since it summarizes the history of possession and therefore the actions performed on the info. Recent analysis [6] highlighted the key contribution of birthplace in systems wherever the utilization of fly-by-night knowledge might result in ruinous failures (e.g., SCADA systems). Though birthplace modeling, collection, and querying are studied extensively for workflows and curated databases [7], [8], birthplace in detector networks has not been properly self-addressed. We tend to investigate the matter of secure and economical birthplace transmission and process for detector networks, and that we use birthplace to discover packet loss attacks staged by malicious detector nodes. In a very multi-hop detector network, knowledge birthplace permits the bachelor's degree to trace the supply and forwarding path of a private knowledge packet. Birthplace should be recorded for every packet; however vital challenges arise because of the tight storage, energy and information measure constraints of detector nodes. Therefore, it's necessary to plot a light-weight birthplace answer with low overhead. Moreover, sensors typically operate in associate untrusted atmosphere, wherever they'll be subject to attacks. Hence, it's necessary to deal with security needs like confidentiality, integrity and freshness of birthplace. Our goal is to style a birthplace cryptography and decryption mechanism that satisfies such security and performance desires.

As we know that existing research is mainly opposed by the concept of separate transmission of data individually and its provenance also individually. In this paper we employ both the things into a single channel, where we also use a single channel or sending both the data as well as provenance. Also the traditional provenance security solutions use the intensively the cryptography and digital signature algorithms, we then employ append based data structures to store the data provenance. In the primitive approach we mainly use the fast message authentication code (MAC) schemes and also the Bloom Filters, which are almost of fixed sized networks that mainly, represents the data provenance.

The main contributions of the proposed thesis is:

a. Initially we concentrate on the problem of secure data provenance transmission in a wireless sensor networks, and in turn used to identify the challenges specific to the above context.

b. Next we mainly concentrated on proposing the cuckoo filter (CF) provenance-encoding scheme.

c. Next we mainly concentrated on the concept of provenance decoding and verification at the base station.

d. Finally we extended our secure provenance encoding scheme and develop a mechanism that detects packet drop attacks.

II. RELATED WORK

In this section we mainly discuss about the data provenance and also some of the data decoding schemes which was used in our current application. Now let us look about that in detail in this current section.

ABOUT DATA PROVENANCE

Data lineage /Data Provenance are a process or a data life cycle that includes mainly the data's origins and where it moves over time [9]. This data provenance mainly describes about what happens for the data when it goes to the diverse processes or internal processes. These processes provide the data analysis It describes what happens to data as it goes through diverse processes. It will give a step-wise explanation about the lost output information during the process of data transmission.Normally this type of data provenance is mainly used by the database systems to address the problem of data lost and also the validations that take place in the database tables [10].

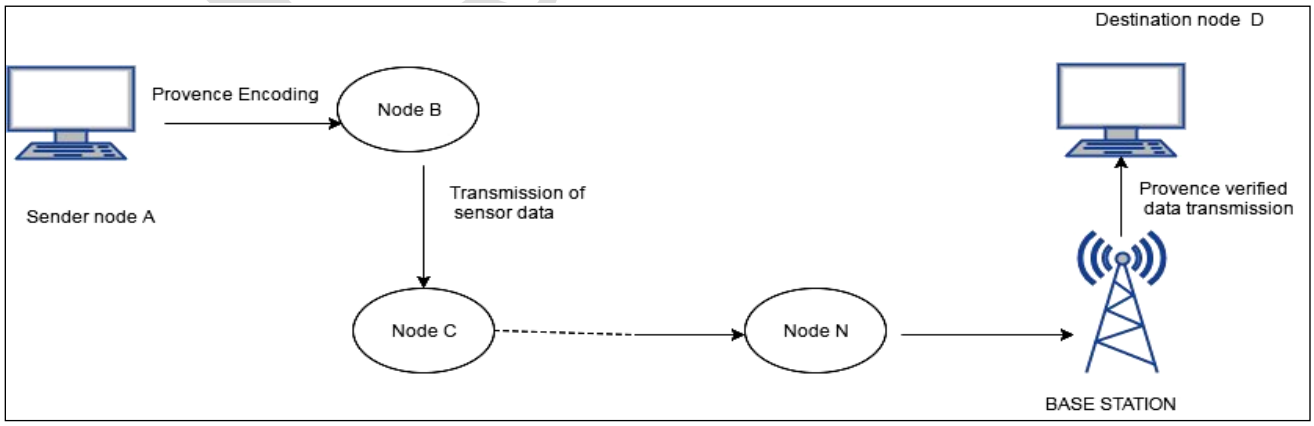

FIGURE 2. REPRESENTS THE FLOW OF PROVENANCE ENCODING AND PROVENANCE VERIFICATION IN A WSN

From the above figure 2, we can clearly identify that this scenario is mainly used for finding the advantage of provenance method during data transmission in a WSN.From the above figure we can clearly find out that there is a single sender node like Node A and there is a single destination node like Node D with a base station which is giving the facility for accessing the information to and from the both the nodes. If the sender wants to send any valuable information to the destination node, it should first pass that information to the various intermediate sensor nodes that are available in the network. Before the data is send to the intermediate nodes the provenance encoding for that data is done at the sender node and then it is forwarded to the intermediate sensor nodes and in-turn the data provenance is verified at the base station and then forwarded the data to the destination nodeD.So by applying the data provenance for the data which is to be transmitted over network, we can clearly find out the exact cause for data loss and data forgery if occurred during data transmission. Here the cuckoo filter with data provenance technique will help the base station to find out the exact cause of packet loss during transmission but they may not optimize or stop the data not to be dropped or forged while transmitting over the network.

THREAD MODEL

In this model we mainly try to find out the very wide range of problems, ranging from simple misconfigurations to hardware faults and even clandestine attacks.We combine assume the overall faults like Byzantine faults [11], i.e., an adversary or intruder may enter into the network illegally and try to compromise an unknown subset of the nodes, and then he try to get the control over them.Thus, the non-malicious problems are covered as a special case in the proposed paper. In this part we assume that the adversary can change both the primary system and the provenance system on these nodes, and he/she can able to read, forge, tamper with, or destroy any information they are holding. We also assume that no nodes or components of the system are inherently safe.

NETWORK MODEL

In this section we mainly assume the network as a multihop wireless sensor network, which consists of a single base station along with a number of sensor nodes and in turn collects the data from the network. The network is modelled as a graph G(N,L), where { Ni | 1≤ i ≤|N| } is the set of nodes, and we assume that L is nothing but the set of links that are available in the graph G.There is also an element like $l_{i,j}$ for each pair of nodes ni and nj that are communicating directly with each other. In this entire application the sensor nodes are always stationary after deployment, but routing paths may change over time, I.e At the time of node failure. Each node tries to report its neighbour node about the information to the Base Station after deployment. The BS assigns each node a unique identifier nodeID and a symmetric cryptographic key Ki. In addition, we try to take a set of hash functions H = {h1, h2, . . . ; hk} in order to broadcast to the nodes for use during provenance embedding.

DATA MODEL

In this data model all the data will be collected in a multi round process from the base station.Initailly each and every sensor node which was available in the network generates data periodically and they finally get aggregated at the Base station level by using some of the existing hierarchical tree based algorithms that are available in the literature [12].Here all the data nodes will be formed as a data path with a set of D hops represented by <N1,N2…N_D >.

Where N1 is assumed as a Leaf Node representing the data source

Here 'i' is the distance between each and every hop starts from leaf node N1.

In this data representation each and every non-leaf node in the path aggregates the received data and provenance with its own locally-generated data and provenance. The following are the common functionalities that an each and every packet contains inside it. They are as follows:

1) A Unique Packet Sequence Number,
2) A Data Value, and
3) A Data Provenance.

For each and every individual packet the sequence number is automatically attached by the data source, and each and every node in the list use the same sequence number for a given round [13]. The sequence number integrity is ensured through MACs, as discussed in later sections.

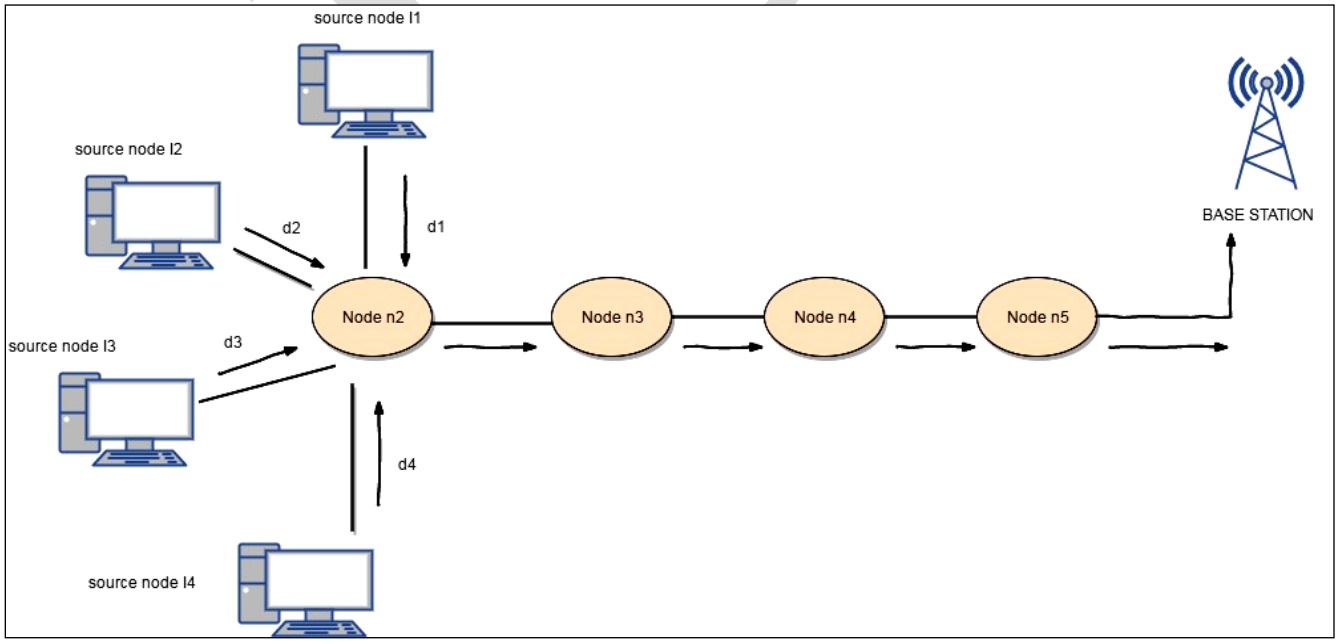

FIGURE 3.REPRESENTS THE FLOW OF DATA IN A WSN THROUGH MULTIPLE SOURCES UNDER A SINGLE BASE STATION

From the above figure 3, we can clearly find out there are multiple source nodes like Source Node1,.,.,.,.,Node5 with a single Base Station. We can also find multiple intermediate sensor nodes termed as node n2,node n3,node n4 and node n5.Here we can observe one thing like each and every individual sender has the ability to send the data through the various intermediate nodes parallel to the base station and in turn can receive the data to the different receivers at same parallel time. Here the provenance corresponding to the packet 'd' is represented as <d1,d2,d3,d4> .The data provenance is the principle which is applied for the proposed application in order to identify the packets loss in terms of packets dropped or packets modified during transmission from a valid source to destination.

III. PROCEDURE FOR DETECTING THE PACKET DROPPING ATTACKS

In this section we will mainly discuss about the proposed procedure for detecting the packet dropping attacks that occur inside the network during data transfer.

MAIN MOTIVATION

The main motivation behind this detection of packet dropping attacks is in the previous section we examined and analyzed the process of secure data provenance encoding techniques with a small sensor network example. Now in this section we try to find out the packet drop attacks that were created by a malicious node inside the network. Initially we assume that links on the current path of our network exhibit the natural packet loss and there may be several adversary nodes that may present in the network during data transmission.

From the below figure 4, we can clearly find out the process of extended provenance encoding process by using Cuckoo filter. If we look at the figure 4 in detail, the data provenance is mainly used to identify each and every record of individual node about the data or packet losses. As we all know that service provider or sender is the one who will try to upload a file into the network. Here the sender will choose only text file as input because it can be divided into packets during transmitting over network.Initailly the service provider or sender node will try to connect with all the intermediate nodes that are available inside the network before sending the data to the valid destination. Here IDS (Intrusion Detection System) acts a major role in maintaining all the log information about the data transmission. If any attack is found during data transmission, it is immediately identified and intimated to the IDS Manager and in turn the packets will be sending to the destination node without any termination.Here the inodes will be used for receiving the packets from previous nodes and in turn forward the packets to the subsequent next nodes until it reaches to the destination node. Here we use CUCKOO filter for filtering the modified and dropped packets and send to the destination node without any delay. Here the CUCKOO filter is mainly used in order to provide countermeasure for the existing filters and send the packets to the destination without terminating the packets at the intermediate inodes. Here in our proposed application our CUCKOO filter is extended in recovering the original file at the receiving node even it was attacked during the intermediate nodes.

.

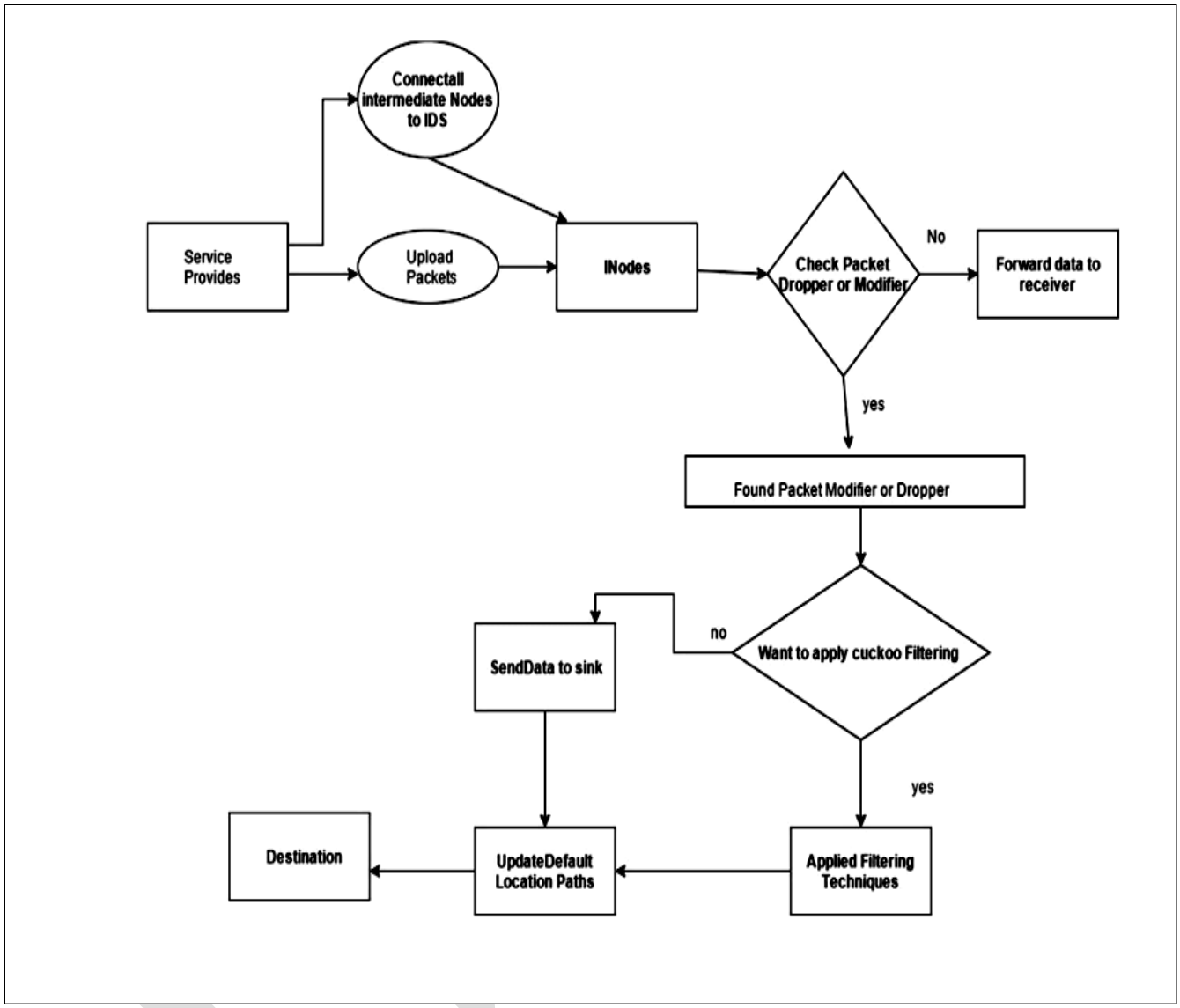

FIGURE 4 .REPRESENTS EXTENDED PROVENANCE FRAMEWORK TO DETECT PACKET DROP ATTACKS AND IDENTIFY MALICIOUS NODES

Initially we try to prove the method of data provenance encoding for the packet acknowledgement that requires the sensors to transmit more meta-data. For each and every data packet ,a provenance record will be generated by a node and it will contains mainly of node ID and an ack for the node in the form of a unique sequence number of the last seen delivered or forward packet. During the process of data transfer from one node to other node if there was any packets dropped due to intermediate nodes failure then we can identify some nodes only can participate in sending the packets from valid source to destination and some are in active state of not carrying any data packets. For this we consider a flow of data oath with term like “P” and n1 is nothing but the data source and we denote the link between the nodes n1 to ni as the li.

IV. CHARACTERISTICS OF CUCKOO FILTER

The main characteristics of a cuckoo filter compared with many previous filters are cuckoo filter has some more advantages than compared with many of the previous filters. In this paper we mainly use the cuckoo filters for encoding the data provenance for the data to be transmitted over a sensor network. Here the main idea behind using the cuckoo filter for data encoding is explained in detail by making comparative analysis with some of the existing filters that are there in the network.

We propose the cuckoo filter, a practical data structure that provides four major advantages.

1. By using the cuckoo filter we can support two functions like adding and removing items dynamically inside the network.
2. Using Cuckoo filter,there was a higher lookup performance than traditional Bloom filters, even when close to full (e.g., 95% space utilized)
3. Cuckoo is easier to implement than alternatives such as the quotient filter; and
4. Cuckoo uses less space than Bloom filters in many practical applications, if the target false positive rate is less than 3%.

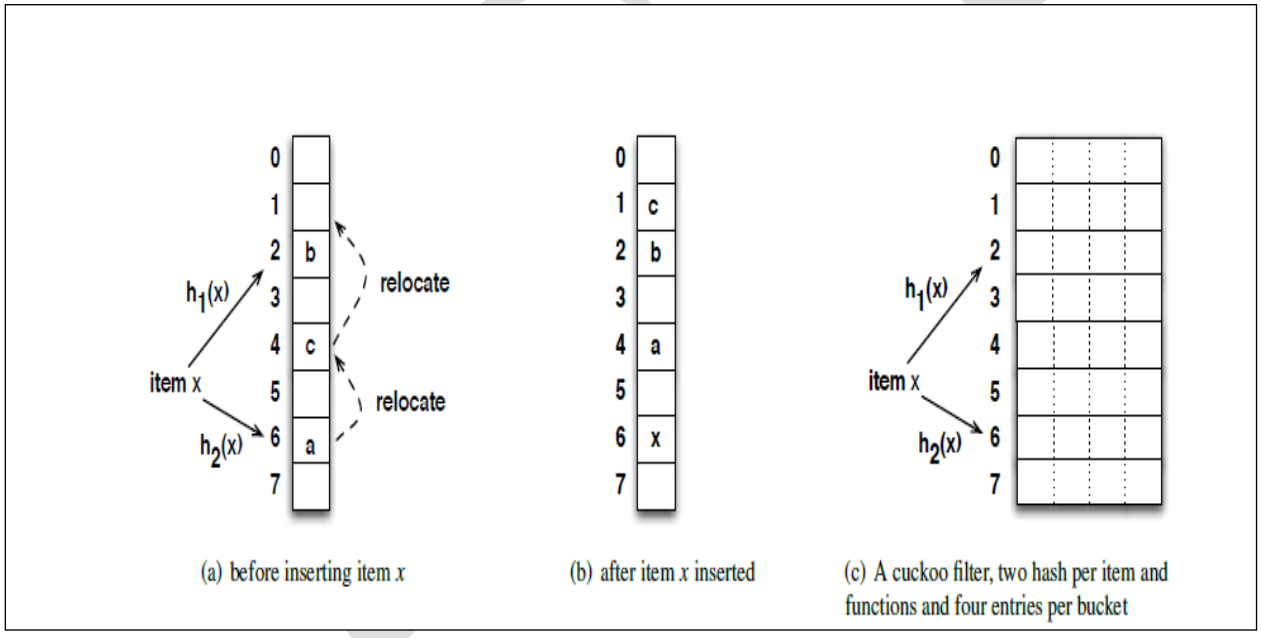

FIGURE 5 .REPRESENTS THE PERFORMANCE EVALUATION OF CUCKOO HASHING

From the above figure 5, we can able to represent the advantage of the basic cuckoo hash table [14].A basic cuckoo has table consists of an array of buckets where each item has two candidate buckets determined by hash functions h1(x) and h2(x). In this we want to perform the lookup process, it then checks both buckets to see whether if there is item available in any of the bucket. Figure 5(a) clearly shows the example of inserting a new item **x** in to a hash table of 8 buckets, where the value 'x' can be placed in either of the buckets like 2 or 6. If there was any

of the two buckets empty ,the cuckoo algorithm try to insert the x to the free bucket that was available in the list and insertion completes. If the same process of insertion occur with neither bucket has space, as is the case in this example, the item selects one of the candidate buckets (e.g., bucket 6), kicks out the existing item (in this case "a") and re-inserts this victim item to its own alternate location. In our example, displacing "a" triggers another relocation that kicks existing item "c" from bucket 4 to bucket 1. This procedure may repeat until a vacant bucket is found as illustrated in Figure 5(b), or until a maximum number of displacements is reached (e.g., 500 times in our implementation). If no vacant bucket is found, this hash table is considered too full to insert [15]. Although cuckoo hashing may execute a sequence of displacements, its amortized insertion time is O (1) [16]-[19].

CUCKOO SEARCH ALGORITHM

The term Cuckoo search is mainly inspired by the brood parasitism of cuckoo species by laying their eggs in the nests of other host birds proposed by Yang and Deb (2009).It is one of the optimization algorithms which are mainly used for inserting, deleting and retrieving data from its buckets by reducing a lot of spaces. For example if we look at a host bird which discover the eggs which is laid in its nest are found as not their own, then it will either throw these foreign eggs away or simply abandon its nest and build a new nest elsewhere. Here we can consider assuming that each egg in a nest as a solution, and a cuckoo egg represents a new solution. The best solution is always replaced with a old solution that was found in the nest. For the simplicity we can consider that each and every net has one egg and each egg can be either best or worst.If a egg is found as best then it will remain same in the nest, but if it found worst or not upto the mark then it will be replaced with a new egg (I.e. With a New Solution). The rules for CUCKOO SEARCH are described as follows

1) Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
2) The best nests with high quality of eggs will carry over to the next generations;
3) The number of available host nests is fixed, and a host can discover an foreign egg with a probability pa∈[0, 1]. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

THE ALGORITHM FOR CS IS GIVEN BELOW:

Generate an initial population of n host nests;

While (t < MaxGeneration) or (Stop Criterion)

Get a CUCKOO randomly (Say i) and replace its solution by levy flights

Evaluate the fitness Fi

Choose a nest among n (Say ,j) randomly;

If(Fi > Fj) ,[for Maximization]

Replace j by new solution;

End if

A fraction (pa) of the worse nests is abandoned and new ones are built;

Keep the best solutions/nests;

Rank the solutions/nests and find the current best;

Pass the current best to the next generation;

end while

V. AES ALGORITHM PSEUDOCODE

As the proposed paper is implemented with a new cryptography technique like encryption and decryption, it is done with the help of AES algorithm. Here in the below section we can clearly get an idea about that AES algorithm in details with the following detailed explanation.

ENCRYPTION PHASE

Step 1: As per the application in the step 1 ,we will upload or browse a file in order to encrypt that before it is stored into the server.

Step 2: **(Encryption of the actual data begins here)**

Let the message to be transmitted be “CRYPTOGRAPHY”.

First find the ASCII equivalent of the above characters.

C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**
67	**82**	**89**	**80**	**84**	**79**	**71**	**82**	**65**	**80**	**72**	**89**

Step 3: Now add these numbers with the digits of the Armstrong number as follows

67 82 89 80 84 79 71 82 65 80 72 89

(+)1 5 3 1 25 9 1 125 27 1 5 3

---

68 87 92 81 109 88 72 207 92 81 77 92

Step 4: Convert the above data into a matrix as follows

A=

$$\begin{bmatrix} 68 & 81 & 72 & 81 \\ 87 & 109 & 207 & 77 \\ 92 & 88 & 92 & 92 \end{bmatrix}$$

Step 5: Consider an encoding matrix...

B = $$\begin{bmatrix} 1 & 5 & 3 \\ 1 & 25 & 9 \\ 1 & 125 & 27 \end{bmatrix}$$

Step 6: After multiplying the two matrices (B X A) we get

C = $$\begin{bmatrix} 779 & 890 & 1383 & 742 \\ 3071 & 3598 & 6075 & 2834 \\ 13427 & 16082 & 28431 & 12190 \end{bmatrix}$$

The encrypted data is...

779, 3071, 13427, 890, 3598, 16082, 1383, 6075, 28431, 742, 2834, 12190

The above values represent the encrypted form of the given message.

DECRYPTION PHASE

Decryption involves the process of getting back the original data using decryption key. The data given by the receiver (the color) is matched [6] with the data stored at the sender's end. For this process the receiver must be aware of his own color being assigned and the key values.

Step 1: As per the application the receiver first chooses the valid data which he want to download it, once he choose that file, he then do the below steps internally in order to view the data in a decrypted manner.

Step 2 :(Decryption of the original data begins here)

The inverse of the encoding matrix is

D = $$(-1/240)*\begin{bmatrix} -450 & 240 & -30 \\ -18 & 24 & -6 \\ 100 & -120 & 20 \end{bmatrix}$$

Step 3: Multiply the decoding matrix with the encrypted data

$$\begin{bmatrix} 68 & 81 & 72 & 81 \\ 87 & 109 & 207 & 77 \\ 92 & 88 & 92 & 92 \end{bmatrix}$$

(D X C) we get

Step 4: Now transform the above result as given below

68 87 92 81 109 88 72 207 92 81 77 92

Step 5: Subtract with the digits of the Armstrong numbers as follows

68 87 92 81 109 88 72 207 92 81 77 92

(-)1 5 3 1 25 9 1 125 27 1 5 3

---

67 82 89 80 84 79 71 82 65 80 72 89

Step 6: Obtain the characters from the above ASCII Equivalent

67 82 89 80 84 79 71 82 65 80 72 89
C R Y P T O G R A P H Y

VI. SYSTEM IMPLEMENTATION

Implementation is the stage where theoretical designs are converted into programmatically manner. Generally in this phase we will try to divide the whole application into several modules and try to analyze each and every module and their individual functionality. In this application we mainly divide the application into four modules, now let us discuss about each and every module in detail. They are as follows:

1. Node Configuration Module
2. Sender Module
3. Router Configuration Module
4. Sink Node Module

Where each and every module again divided into several parts and now let us discuss about each and every thing in detail as follows:

1) NODE CONFIGURATION MODULE

In this module initially we will try to create a set of nodes one with other for data transmission from a valid source node to the destination node. In this module we will assume a single source node and multiple destination nodes along with a set of intermediate nodes like INode1, INode2 and so on till INode4.Where each and every intermediate nodes are meant for accepting the packets from the previous node and send the data to the consequent next nodes and in turn passes the data finally to the destination. We create the network group by connecting nodes to sink. Link configuration means connecting the nodes and intermediate nodes to the sink.

2) SENDER MODULE

This sender module is again divided into two sub parts where each and every part has individual functionality. Now look at those individual sub modules as follows:

a) PACKET SPLITTING

In this module, the source node or sender node try to select a text file which is to be sent to the valid destination. Initially after choosing a valid text file, he then try to split the file into number of parts I.e like ten parts or ten packets with equal sized packets. If during the division of packets if there is any size variance then adding of bits at the padding field is applied in order to make that packet into a meaningful way.

b) SEND PACKETS TO INTERMEDIATE NODES

Once the sender splits the file into individual packets, he then try to encrypt the individual splitter packets and then try to add some bits at the padding field and try to send the packets to the valid destination nodes. Here each and every bit is added by the sender to make the packets un-identified by the un-authorized users who wish to view the data during data transmission. After adding of bits to each packet, it sends the packets to the nearest node or intermediate node.

3) ROUTER INITIALIZATION MODULE

In this router initialization module all the intermediate nodes like INode1,INode2…INode4 are initialized and they are kept ready for receiving the packets to and from the previous nodes and send the packets successfully to the next node and finally transmit the data to the destination node. Again this router module is sub-divided into several individual modules; now let us look at those individual modules in detail as follows:

a) SEND PACKETS TO SINK NODE

In this module, the intermediate node receives Packets from the sender. After receiving all packets from sender, it encrypts all packets again for authentication. Before sending to sink, intermediate add some bits to each packet for node identification. After adding some bits from intermediate, it sends all packets to the sink.

b) MODIFY OR DROP

Before sending all packets to sink, packets dropping or packets modifying may be occur in intermediate.

4) SINK NODE MODULE

This module is the last module for our proposed application where this sink node is nothing but the receiver who try to wait for accessing the data which is send from the valid source node .Again this sink node is divided into three more sub modules ,they are as follows**:**

a) VERIFY MODULE

In this module, Sink receives all packets from the sender node, and it verifies all packets which are dropped or not. And it also verifies the packets which are modified or not and it can identify the modifiers in the process based on the bit identification.

b) MERGE PACKETS MODULE

After receiving all packets in sink, it decrypts all packets. After the decryption if there is no modified or dropped packets, it merge all packets. After merging, Sink can receive the original file.

c) CATEGORIZATION AND RANKING MODULE

In this module Categorization and Ranking will be performed based on the node behavior. If there is any modification or drop of packets in node it assumes negative value for modifier or dropper. Sink performs Ranking for each node based on the Category of nodes. Sink gives ranking like Good, Temporarily Good, Suspiciously Bad, Bad based on the node behavior in the process

VII.CONCLUSION

We finally proposed a novel light-weight secure provenance encoding and decoding scheme based on cuckoo filters. This scheme is mainly used for ensuring the confidentiality, and data integrity of the data provenance. Also as an extension we also included a facility to find or detect the packet dropping attacks that was created by a hacker or intruder in the network. This was mainly monitored by the router in order to find the packet droppers who are available within the network during data transmission. By conducting various experiments on our proposed scheme, the proposed cuckoo filter for encoding the data provenance gives best result in terms of effectiveness, scalability and even it is light weight for retrieval of data in a short period of time. As a future work we plan to implement the same concept on a multiple consecutive malicious sensor nodes for identifying the attackers as the current application is limited up to single malicious sensor node.

VIII. REFERENCES

I. Vaudenay, Serge (September 16, 2005). A Classical Introduction to Cryptography: Applications for Communications Security (1st ed.)

II. "Understanding Denial-of-Service Attacks". *US-CERT. 6 February 2013*. Retrieved 26 May 2016.

III. *Prince, Matthew (25 April 2016).* "Empty DDoS Threats: Meet the Armada Collective". *CloudFlare*. Retrieved 18 May 2016.

IV. "Brand.com President Mike Zammuto Reveals Blackmail Attempt". *5 March 2014. Archived from* the original *on 11 March 2014.*

V. "The Philosophy of Anonymous". *Radicalphilosophy.com. 2010-12-17*. Retrieved 2013-09-10.

VI. H. Lim, Y. Moon, and E. Bertino, "Provenance-Based Trustworthiness Assessment in Sensor Networks," Proc. Seventh Int'l Workshop Data Management for Sensor Networks, pp. 2-7, 2010.

VII. I. Foster, J. Vockler, M. Wilde, and Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation," Proc. Conf. Scientific and Statistical Database Management,pp. 37-46, 2002.

VIII. K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, "Provenance-Aware Storage systems," Proc. USENIX Ann. Technical Conf., pp. 4-4, 2006.

IX. http://www.techopedia.com/definition/28040/data-lineage

X. De, Soumyarupa. (2012). Newt : an architecture for lineage based replay and debugging in DISC systems. UC San Diego: b7355202. Retrieved from: https://escholarship.org/uc/item/3170p7zn.

XI. P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire Tinyos Applications," Proc. Int'l Conf. Embedded Networked Sensor Systems, pp. 126-137, 2003.

XII. S. Madden, J. Franklin, J. Hellerstein, and W. Hong, "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks," ACM SIGOPS Operating Systems Rev., vol. 36, no. SI, pp. 131-146, Dec. 2002.

XIII. K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An Efficient Clustering Based Heuristic for Data Gathering and Aggregation in Sensor Networks," Proc. Wireless Comm. and Networking Conf., pp. 1948-1953, 2003.

XIV. F. Putze, P. Sanders, and S. Johannes. Cache-, hash- and spacee_cient bloom filters. In Experimental Algorithms, pages 108–121. Springer Berlin / Heidelberg, 2007.

XV. A. Broder, M. Mitzenmacher, and A. Broder. Network Applications of Bloom Filters: A Survey. In Internet Mathematics, volume 1, pages 636–646, 2002.

XVI. E. Perla, A. Cath_ain, R.S. Carbajo, M. Huggard, and C.M. Goldrick, "Powertossim z: Realistic Energy Modelling for Wireless Sensor Network Environments," Proc. ACM Workshop Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, pp. 35-42, 2008.

XVII. W. Zhou, Q. Fei, A. Narayan, A. Haeberlen, B. Loo, and M. Sherr, "Secure Network Provenance," Proc. ACMSOSP, pp. 295-310, 2011.

XVIII. A. Ramachandran, K. Bhandankar, M. Tariq, and N. Feamster, "Packets with Provenance," Technical Report GT-CS-08-02, Georgia Tech, 2008.

XIX. W. Zhou, M. Sherr, T. Tao, X. Li, B. Loo, and Y. Mao, "Efficient Querying and Maintenance of Network Provenance at Internet-Scale," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 615-626, 2010.

IX. ABOUT THE AUTHORS

KONA DIVYA is currently pursuing her 2 years M.Tech in Department of Computer Science and Engineering at Pydah College of Engineering and Technology, affiliated to JNTUK University, AP, India. Her area of interest includes Networks and Security.

A.V.D.N.MURTHY completed his MCA and M.Tech. in Computer Science and Engineering. He is currently working as Assistant Professor of Department of Computer Science and Engineering at Pydah College of Engineering and Technology, affiliated to JNTUK University. He is having industrial experience of 1.2 years and teaching experience of 11 years. His areas of interest include Data mining, Image Processing, Cryptography & Network security, Computer Networks and Operating Systems.