Abstract—The work we have done after the project proposal can be summarized as follows. We revised our CATCH scheme based on the feedbacks from Dr. Wallach and an anonymous reviewer and added more details of the CATCH scheme, in particular the proof construction and packet-dropping-investigation mechanism. We also implemented a part of CATCH scheme on ns2 network simulator. This report gives the details of these work.

I. CATCH SCHEME OVERVIEW

This section overviews the CATCH scheme we design to identify packet droppers in wireless ad-hoc network. Instead of repeating the content in our project proposal, we only highlight the important details that answer the questions of the project proposal reviewers.

CATCH is a fully distributed scheme in that using CATCH, every node independently detects whether its packets are dropped and launches investigations to identify the packet droppers on the routes of the packets sent.

A node in CATCH stores a proof for a data packet it forwarded. The proof of a packet is obtained from the packet receiver and is constructed using symmetric cryptography, which is much faster than asymmetric cryptography. CATCH requires the data packet destination to provide an ACK packet for the data packet received, which enables the source of the data packet to know whether a data packet is received. If a source node detects that a fraction of its packets are not received by the destinations, it launches an investigation for a lost packet and requests every node on the route of the lost packet to present the proof of forwarding the lost packet.

A node $S$ in CATCH maintains the packet-dropping metric for every node that is on the routes of the packets generated by $S$. $S$ increases the packet-dropping metric of a node $X$ if $X$ is unable to present a proof for the packet under investigation. The details of computing the packet-dropping metric are given in [11].

The malicious nodes may forge a great number of investigation requests to consume the network bandwidth and the system resource of other nodes. To defend such attack, a node discards the investigation requests for the packets it did not forward before. Hence, the large number of fake investigation requests from a malicious node essentially become wireless interference to its neighbors.

The packet dropper information known by one node can potentially help other nodes to decide which nodes are packet droppers. However, in current design of CATCH, a node decides who are packet droppers based on its own investigations and does not share the packet dropper information with another node. As a node does not know which nodes are trustworthy, this design prevent the malicious nodes to disseminate wrong packet dropper information to other nodes.

Next we present new details of the CATCH scheme.

II. SYSTEM MODEL

A. Adversarial Model and System Assumptions

The adversarial model in this project is that a malicious node probabilistically drops packets it forwards. CATCH can also discover the malicious nodes that drop all packets going through them. The assumptions in CATCH protocol are listed as follows.

1) Two nodes in the network share a pairwise symmetric key so the messages between them can be encrypted using symmetric cryptography.

2) All wireless links in the network are symmetric; that is, node A can hear node B implies node B can hear node A.

3) In the network, the malicious nodes are assumed to account for minority. The packet transmission and forwarding is based on source routing so the packet source knows the route of a packet. DSR is used as the routing protocol in this work.

The notations used in this article are defined as follows.

- $S$: the message source.
- $D$: the message destination.
- $SK_{a,b}$: the symmetric secret key between node $a$ and node $b$.
- $ES_{SK_{a,b}}(M)$: the encryption of a message $M$ using the shared secret key of node $a$ and node $b$.
- $E^{-1}_{SK_{a,b}}(M)$: the decryption of a ciphertext $M$ using the shared secret key of node $a$ and node $b$.

B. Packet Transmission Proof

A wireless mobile ad-hoc network (MANET) consists of nodes that are wirelessly connected to each other and a packet from a source is forwarded by the intermediate nodes until reaching the destination. Figure 1 illustrates a packet transmission in a MANET.

A packet from the source contains the route from the source to the destination. The route is in plain text but the message in the packet is encrypted. The source generates a message authentication code (MAC) for the packet using the shared secret key between the source and the destination and SHA1 hash function. Furthermore, for each intermediate node $I$ on the route, the source generates a MAC for the route and the sequence number ($seq$) using the shared secret key between the source and $I$. Thus, each node on the route can verify whether
the route and packet sequence number have been modified. A packet \( P \) with content \( M \) is sent in the following format:

\[
P = \{S, D, seq, Route, MACs, E_{SK_{S,D}}(M)\}
\]

The MAC that \( S \) generates for a node \( I \) on the route is:

\[
MAC(M) = \{E_{SK_{S,I}}(SHA1(S||D||seq||Route||M))\},
\]

in which \( M \) denotes the message in the packet. Since a malicious node does not know \( SK_{S,I} \), if it modifies any part of the message \( M \), the receiver of the MAC(M) will discover that the MAC(M) does not match the modified packet.

When a node \( I \) receives a packet \( P \), it verifies the validity of the route and the message by checking that the MAC in the packet is valid. If the route or the message are invalid, node \( I \) will discard the packet. If the route and the message is valid, node \( I \) provides a packet transmission proof to upstream and forwards the packet to the downstream node on the route. If the message destination receives the packet, it verifies the MAC for the packet and sends an ACK packet to the source to indicate that the packet is received.

The proof sent to \( I \) from the downstream hop \( J \) is in the following format. The \( seq \) in the proof denotes the packet sequence number in the received packet.

\[
Proof = \{S, J, seq, Route, MACs, E_{SK_{S,J}}(S,D,I,seq)\}
\]

The \( E_{SK_{S,J}}(S,D,I,seq) \) in the proof indicates that \( J \) has received the packet from \( I \) and the other fields in the proof record the packet information corresponding to this proof.

C. Investigation

Based on the number of data packets generated and the number of ACKs it receives, a source node can detect whether the packet loss rate is higher than a threshold. If so, it would hold an investigation to find out where the packets are dropped or lost, and then the nodes near the drop point would be regarded as bad or malicious nodes. In the investigation, the source node looks through the proofs from the nodes on the route. If \( S \) did not receive the proof from \( I \) but receive the proof from a downstream node of \( I \) (i.e., a node nearer to the destination than \( I \)), then \( I \) is not regarded as malicious node. If \( S \) did not receive the proof from \( I \) and the proof from a downstream node of \( I \), then the packet dropping metric of \( I \) is incremented by \( S \).

Similarly, \( D \) and intermediate nodes investigate the nodes that may drop the ACK and proof packets. Since the packet dropper \( X \) is unable to distinguish the data packet, the ACK packet, and the proof packets. The packet dropping metric of \( X \) will be higher than that of other nodes since \( X \) is unable to obtain the proofs for the packets it dropped.

The investigation algorithm is as follows.

1) Based on the route of a packet \( P \), \( S \) checks the proof received from every node on the route, starting from the node that is the immediate downstream node of \( S \).

2) If the proof from a node \( I \) is not received, check if the proof from a downstream node of \( I \) has been received. If no proof from a downstream node of \( I \), then the packet dropping metric of \( I \) and its predecessor is incremented and investigation stops.

3) If the proof from every node except the destination has been received, \( S \) stops the investigation as this indicates that the packet from \( S \) has reached the destination and the ACK packet was lost.

Based on the investigation mechanism, the source node can narrow down the scope of suspects to two nodes. For instance, node \( S \) sends packets to the destination node \( D \), and these packets are forwarded by \( A \), \( B \), and \( C \) in order. If a packet is lost and \( A \) provides the proof of sending to \( B \) but \( B \) cannot provide any proof of sending to \( C \), then \( S \) would consider that the packet is lost/dropped between \( B \) and \( C \).

III. THE PACKET-DROPPING METRIC

A. Basic concept

A node \( S \) constructs a packet-dropping metric for each node that is on the routes of the packets sent by \( S \), which measures how likely a node is a packet dropper. When \( S \) considers another node dropped a packet generated by \( S \), \( S \) increments the packet-dropping metric for this node. On the other hand, after a few successful packet deliveries, \( S \) decreases the packet-dropping metric of the nodes on the routing path. A node with high packet-dropping metric would be considered as a malicious node or bad node, i.e., a node running out of power or getting high interference nearby. If possible, a source node would avoid selecting a node with high metric when deciding the routes.

B. Details

In order to maintain the metrics, a source node must create a mapping between identities of nodes and their corresponding metrics. In real world, the metric of an identity is just like the credit score under a specific SSN. Here we make an assumption that each node in the MANET has a unique identity, which is invariable, and each node can know other nodes’ identity easily. For simplicity, we use IP address as the identity of a node and assume nodes would never change their IP addresses in this work.
At the first time a source node $S$ sees another node $N$ in its route, $S$ would assign an initial metric $m_{\text{init}}$ to $N$ and add this record in its metric table. The metric of a node cannot exceed the maximum $m_{\text{max}}$ or be lower than the minimum $m_{\text{min}}$. The metric is updated based on the following EWMA formula:

$$\text{metric} = \text{metric} \times (1 - \alpha) + (\text{metric change}) \times \alpha,$$

where $\alpha$ is the constant smoothing factor between 0 and 1, and it stands for the importance of the latest observation. On a successful packet delivery, a node’s packet-dropping metric would be decreased; that is, the value of $\text{metric change}$ would be negative. On the other hand, $\text{metric change}$ would be positive under suspicious packet dropping.

If a node’s packet-dropping metric is lower than a threshold $m_{\text{low}}$, the source will consider to remove this “bad” node from its routes and discover new routes if necessary. Afterwards, the metrics of all the downstream nodes of this removed node in the original routes would be reset to the average metric of other nodes in the routes. This prevents the innocent downstream nodes from being judged as bad in a short time.

If a node’s packet-dropping metric has not been updated for a period of time, say 5 minutes, its metric would be gradually (i.e., every minutes) increased to $m_{\text{init}}$. This gives the removed node a chance if it is no longer malicious or its batteries have been replaced.

### IV. Implementation on Ns-2 Network Simulator

We have implemented a part of CATCH system on the ns-2 network simulator, including the implementation in MAC layer to construct packet transmission proofs, and the modification of DSR code in network layer to send packets with sequence numbers and request the destinations to send ACK packets.

Next we will finish implementing the CATCH protocol. In order to evaluate the correctness and effectiveness of the CATCH scheme, we plan to measure the packet-dropping metric of a node and show its relationship with the packet drop rate of malicious nodes. Next, we plan to evaluate the performance of our system under different mobility models. The performance metrics include the packet delivery ratio (PDR), the false positive rate (FPR) and false negative rate (FNR) of malicious node detection, and the overhead of the proof messages.