Mitigating Stealthy Collision Attack in Energy Harvesting Motivated Networks

Cong Pu[†] Sunho Lim[§] Byungkwan Jung[§] Manki Min[‡]
[†]Division of CS, Marshall University, Huntington, WV 25755, puc@marshall.edu
[§]Dept. of CS, Texas Tech University, Lubbock, TX 79409, {sunho.lim, byung.jung}@ttu.edu
[‡]Dept. of EECS, South Dakota State University, Brookings, SD 57007, manki.min@sdstate.edu

***Abstract*—Energy harvesting motivated networks (EHNets) are rapidly emerging as a major part of ubiquitous computing and communication infrastructure in the presence of Internet-of-Things (IoT). A set of self-sustainable nodes equipped with energy harvesting capabilities can effectively exploit ambient energy and convert it into electric energy, but it is admittedly vulnerable to a Denial-of-Service (DoS) attack that primarily targets service availability, often witnessed in wireless multi-hop networks. In this paper, we propose an adaptive acknowledgment-based approach, called AAA, to detect the stealthy collision attack caused by multiple malicious nodes in the realm of EHNets, where two malicious nodes coordinate their packet transmissions simultaneously to create the packet collision at a legitimate node. In the AAA, each node forwards a *Data* packet, monitors the subsequent packet transmission of its one-hop downstream node and waits for an explicit acknowledgment (*Ack*) packet from its two-hop downstream node, and then detects the stealthy collision attack in EHNets. We conduct extensive simulation experiments using OMNeT++ for performance evaluation and comparison. The simulation results indicate that the proposed countermeasure can provide higher detection rate and packet delivery ratio but lower detection latency compared to the existing scheme, MCC.**

***Index Terms*—Acknowledgment-based detection, colluding collision, energy harvesting motivated networks.**

I. Introduction

Internet-of-Things (IoT) and its applications are rapidly proliferating, where a myriad of multi-scale sensors and devices (later in short, nodes) are seamlessly blended [1]. Nodes are resource constrained in terms of computing, storage, and battery power, but they are often required to operate a long-term sensing and communicating in a remote or unattended area. Due to the limited battery power, it is ultimately unavoidable to replace or replenish batteries. Energy harvesting has emerged as a promising technology to extend the lifetime of nodes by continuously harvesting environmental resources, such as sunlight, wind, vibration, etc. This paper is also motivated by the fact that the U.S. Army planned to eliminate all the military batteries or at least reduce the frequency of replacing batteries for communication devices [2]. Soldiers will be equipped with battery-less or self-powered communication devices [3]. We envision that an energy harvesting motivated network (EHNet) will be a major part of ubiquitous computing and communication infrastructure in IoT, where a set of self-sustainable nodes equipped with energy harvesting capabilities communicate directly or indirectly via multi-hop relays. However, EHNet is indeed vulnerable to Denial-of-Service (DoS) attacks [4], primarily targeting service availability, due to the lack of centralized coordination, physical protection, and security requirement, often witnessed in wireless multi-hop networks.

In this paper, we investigate stealthy collision attack and its countermeasure in EHNets, where multiple malicious nodes collude together to create packet collisions at a legitimate node on purpose to drop the packet without being detected. It is not trivial to identify this intentional packet drop from accidental packet collisions. Note that this is different from selective forwarding attack [4], where a malicious node randomly or strategically drops incoming packets. Countering stealthy collision attack and its variants in battery-powered networks have been studied in [5], [6]. Unfortunately, stealthy collision attack and its countermeasure are still in its infancy and under-explored in the realm of EHNets. In light of this, we propose an adaptive acknowledgment-based approach, called *AAA*, to efficiently detect the stealthy collision attack of multiple malicious nodes in EHNets, where each node periodically harvests energy and repeats on- and off-period for communication. Our major contributions are two-fold:

- First, we analyze a camouflage-based active detection [7] and identify its vulnerability in which the adversary can launch a stealthy collision attack. We measure the impact of stealthy collision attack on packet delivery ratio (PDR) as a preliminary result.
- Second, we propose an adaptive acknowledgment-based approach, called *AAA*, to efficiently detect the stealthy collision attack caused by multiple malicious nodes. In the AAA, each node forwards a *Data* packet, monitors the subsequent packet transmission of its one-hop downstream node, and waits for an explicit acknowledgment (*Ack*) packet from its two-hop downstream node.

We develop a customized simulation framework using OMNeT++ [8] to conduct the performance evaluation study in terms of five performance metrics and show the viability of the proposed approach to stealthy collision attack in EHNets.

The rest of paper is organized as follows. Prior schemes are summarized and analyzed in Section II. The system and adversarial models and the proposed countermeasure are presented in Sections III and IV, respectively. Extensive simulation experiments and their results are presented in Section V. Finally, concluding remarks are provided in Section VI.

978-1-5386-0595-0/17/$31.00 ©2017 IEEE

II. Related Work

Both watchdog and pathrater techniques [9] are designed to detect and mitigate routing misbehaviors. A watchdog technique detects a misbehaving node by monitoring its transmission to see whether it forwards a packet within the requisite delay bound without modification and fabrication. A node is suspected as a malicious node, if a certain number of neighboring nodes report the node as a misbehaving node. [5] and [6] are a variant of [9], where the monitoring nodes are extended from the common neighbors of the packet sender and the forwarding node to all the neighbors of the forwarding node. Each node records the number of forwarded packets for its adjacent node, and the node is suspected as a malicious node if it shows two different views of the volume of the forwarded traffic.

In [10], the 2ACK is proposed to detect misbehaving links in mobile ad hoc networks (MANETs), where each intermediate node located along the forwarding path generates an *Ack* packet and forwards it to a two-hop neighbor node in the opposite direction of the data traffic after receiving the data packet. Each node observes the behavior of link by recording the number of received *Ack* packets for a certain time period. If the link shows a higher *Ack* packet loss ratio than a pre-defined threshold value, this link is declared as a misbehaving link and added to the blacklist. In SCAD [11], a single checkpoint-assisted approach integrated with timeout and hop-by-hop retransmission techniques is proposed to detect a selective forwarding attack in wireless sensor networks (WSNs), where single or multiple malicious nodes randomly or selectively drop any incoming packet.

In CRS-A [12], each node maintains a reputation table with adaptive detection threshold to evaluate the forwarding behavior of its adjacent nodes in WSNs. The reputation value is calculated based on the deviation of the monitored packet loss rate as well as the estimated normal loss rate caused by the time- and location-variant channel quality and the link layer collisions. The node with low reputation value is detected and isolated from the routing path.

A hop-by-hop detection scheme [13] is proposed to detect the forwarding misbehavior in energy harvesting motivated WSNs. In [13], each node records the trace of forwarding operations using overhearing and exchanges the trace information with its adjacent nodes to detect any forwarding misbehavior. When a node detects a forwarding misbehavior, it reduces a forwarding probability of the suspected node. In [7], a camouflage-based approach is proposed to detect the stealthy selective forwarding attack in EHNets. Each node actively disguises itself as an energy harvesting node on purpose and pretends not to overhear, and then monitors any forwarding operation of its adjacent nodes to detect a lurking malicious node.

In summary, stealthy collision attack and its variant forwarding misbehaviors have been well studied primarily in battery-powered networks. However, little attention has been paid for energy harvesting enabled devices in the realm of EHNets.

III. System and Adversarial Models

First, we consider an energy harvesting motivated network, where each node is assumed to equip with an energy harvester and a rechargeable battery. For example, a piezoelectric fiber composite has been utilized to transduce mechanical vibration energy into electrical energy [14]. A piezoelectric fiber composite bimorph (PFCB) W14 can generate about 1.3 mW to 47.7 mW, which is sufficient for the communication activities of most small wireless sensors [15]. Due to the nature of intermittently available harvesting resource, energy harvesting process is modeled as a two-state Markov process with active (s_a) and harvest (s_h) states. Each node stays in either active or harvest state for a certain time period, which is exponentially distributed with a mean λ_a or λ_h respectively, and switches between states. To avoid energy consumption and operational delay of frequent state changes, we adopt the *charge-and-spend* energy harvesting policy, where a node in harvest state is unable to listen and receive any packet until a certain level of energy is harvested. During a network deployment phase, each node exchanges a one-time single-hop *Hello* packet to build a list of neighbor nodes. When a node detects an event, it becomes a source node, generates a *Data* packet, and forwards the packet toward a sink. To deliver the packet to the sink, lightweight forwarding protocols [16] can be deployed.

Second, the primary goal of adversary is to reduce the network performance by interrupting on-going communication. The adversary is able to capture and compromise legitimate nodes so that they can behave maliciously. A malicious node may eavesdrop any on-flying packet and inject false information or modify the packet to mislead the network traffic on purpose. However, if a packet sender can authenticate a packet with a lightweight digital signature [17], a receiver can easily verify the packet and detect any modification. We assume that the malicious node has no constraint in terms of energy and memory. Here, we consider a dense network where multiple forwarding candidate nodes are available. Thus, sub-networks connected by a single node is not considered because it could be a single point of failure or a malicious node. In this paper, we focus on the adversarial scenarios that cannot be detected by digital signature and cryptographic techniques. We do not consider cryptographic primitives.

IV. Mitigating Stealthy Collision Attack

In this section, we investigate the stealthy collision attack with a preliminary result and propose an adaptive acknowledgment-based approach, called *AAA*, to efficiently detect the stealthy collision attack caused by multiple malicious nodes in EHNets.

A. Potential Vulnerability and Implication

In [7], a camouflage-based active detection (CAM) is proposed to detect selective forwarding attack in EHNets. In order to prevent node from mistakenly forwarding a packet to a node in harvest state, causing unexpected packet losses, each node in harvest state periodically broadcasts a one-hop *State* packet. When a node harvests enough energy and switches to

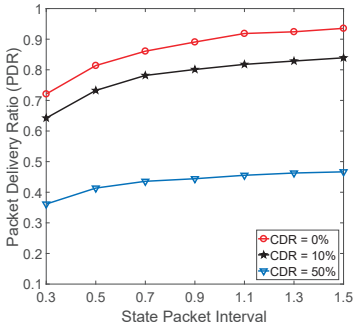


Fig. 1. The performance of PDR against both *State* packet interval and collaborative drop rate (CDR).

active state, it broadcasts a one-time *State* packet. The node does not periodically broadcast the *State* packet in active state. The *State* packet interval is uniformly distributed with a mean φ_{sp}, e.g., φ_{sp} = 1.5 (sec). In Fig. 1, we observe the impact of *State* packet intervals with uniform distribution on packet delivery ratio (PDR) without malicious node, denoted as CDR = 0%. Low PDR is observed with short *State* packet interval because frequently broadcasted *State* packet can be collided with *Data* packet. As *State* packet interval increases, PDR increases. More than 92% PDR is observed when the interval is close to 1.5 (sec).

However, attacker may exploit the existence of *State* packet to launch stealthy collision attack that leads to packet drop without being detected. For the sake of simplicity, we use a snapshot of network consisting of seven energy harvesting nodes as shown in Fig. 2. A packet sender n_a forwards a *Data* packet to node n_d through one of forwarding candidate nodes (n_b, n_c or n_{m1}). Suppose n_{m1} and n_{m2} are malicious nodes and they can unrestrictedly switch the state between active and harvest. When n_a is in active state and has a *Data* packet to send, it randomly selects one of forwarding candidate nodes and sends the packet. In Subfig. 2(a), suppose n_a selects n_{m1} as a forwarding node and sends the *Data* packet. Since n_b and n_c are in active state, they can overhear the packet forwarding and then store the *Data* packet in their local cache. If n_{m1} behaves normally and forwards the packet to n_d, then n_a, n_b, n_c and n_e can overhear the packet and assume that the packet has been successfully forwarded to the next hop node, n_d.

But n_{m1} and n_{m2} may collude together and coordinate the *Data* packet and *State* packet transmission to drop the packet without being detected. For example, n_{m2} intentionally switches to harvest state and periodically broadcasts a *State* packet. n_{m1} can coordinate the *Data* packet transmission to n_d with the broadcasted *State* packet from n_{m2}. This simultaneous transmission causes a packet collision at n_d, which results in packet loss. Both n_a and n_c can overhear the packet forwarding and assume that the packet has been successfully forwarded to the next hop node. n_e overhears the packet transmission and stores the packet in its local cache. However, n_e will not be able to overhear the packet forwarding from n_d. When the timeout period expires, n_e will suspect the forwarding misbehavior of n_d. On the other

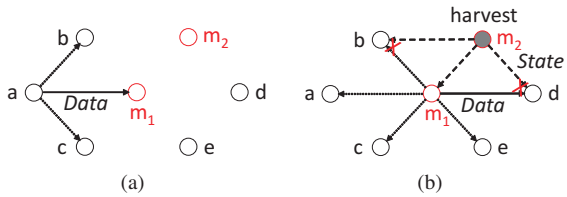


Fig. 2. Packet drop through stealthy collision attack. Here, a malicious node and a node in harvest state are marked as red and shade, respectively. Solid, dotted, and dash lines represent forwarding, overhearing, and periodic broadcast operations, respectively.

side, n_b cannot overhear the packet transmission due to the collision of *Data* and *State* packets either. After the timeout period, n_b will suspect the forwarding misbehavior of n_{m1}. However, this suspicion will be considered as a bad mouth attack [18], because the majority of neighbor nodes (i.e., n_a and n_c) successfully overhear the packet transmission.

Based on the aforementioned adversarial scenario, we measure PDR with different collaborative drop rates (CDR) of multiple malicious nodes in Fig. 1, denoted as CDR = 10% and 50%. The overall PDR with CDR = 10% and 50% are lower than that of PDR with CDR = 0%, because multiple malicious nodes collude together to launch stealthy collision attack, resulting in packet losses. As the *State* packet interval increases, the PDR increases slightly. This is because the malicious nodes intentionally drop the *Data* packet through collision, the less number of *Data* packet reaches the destination. When CDR = 50%, PDR significantly drops below 50%.

B. Adaptive Acknowledgment-based Approach

The basic idea of the proposed approach is that each node forwards a *Data* packet, and then monitors the subsequent packet transmission of its one-hop downstream node and waits for an explicit acknowledgment (*Ack*) packet from its two-hop downstream node, respectively. If the packet sender does not overhear the *Data* packet forwarded by its one-hop downstream node or receive the *Ack* packet from its two-hop downstream node before the timeout period, it suspects the forwarding misbehavior of one-hop downstream node.

First, when a node receives a *Data* packet, it randomly selects one of the active one-hop downstream nodes as a forwarding node. If none of the forwarding nodes is in active state, the node replies a *Hold* packet to the packet sender and caches the *Data* packet in its local storage. When the node overhears a *State* packet from an active one-hop downstream node, it forwards the cached *Data* packet. When a node forwards a *Data* packet, it records the number of forwarded *Data* packet (N^{fwd}). It also decides whether to request the two-hop downstream node to reply an *Ack* packet. We deploy an acknowledgment probability (P^{ack}) that indicates how frequently the packet sender requests two-hop downstream node to reply the *Ack* packet. P^{ack} is adaptively adjusted based on the number of received *Ack* packets. If a random number (e.g., rand[0,1]) generated by the packet sender is less than or equal to P^{ack}, it adds the *id* of the forwarding node in the list of packet sender of acknowledgment node, *LP*,

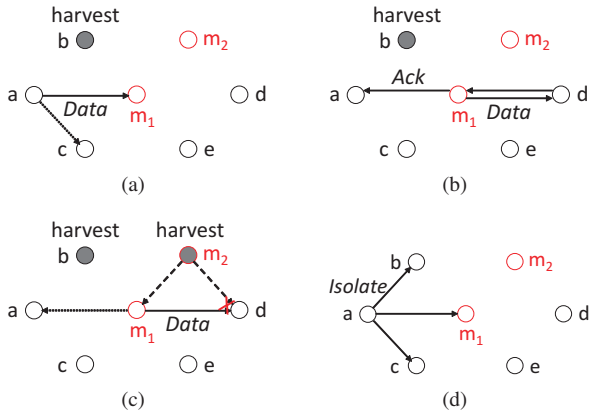


Fig. 3. A set of snapshots of the proposed AAA scheme.

which is piggybacked in the *Data* packet, and then increases the number of requested *Ack* packet (N^{ack}) by one. When a node receives the *Data* packet, it checks whether the packet sender's *id* is in the *LP*. If the packet sender's *id* is in the *LP*, the node forwards the *Data* packet to one of active one-hop downstream nodes and replies an *Ack* packet back to two-hop upstream node. For example, in Subfig. 3(a), suppose node n_a selects n_{m1} as the forwarding node and decides to request the two-hop downstream node to reply the *Ack* packet (e.g., $P^{ack}_{m1} \leq$ rand[0,1]). n_a adds the *id* of n_{m1} in the *LP*, *LP* = *LP* $\cup$ $[m1]$, piggybacks the *LP* in the *Data* packet, increases N^{fwd}_{m1} and N^{ack}_{m1} by one respectively, and then forwards the *Data* packet to n_{m1}. As shown in Subfig. 3(b), when n_d receives the *Data* packet forwarded by n_{m1}, it checks that the *id* of packet sender n_{m1} is in the *LP*. Thus, n_d forwards the *Data* packet to next hop node and then generates an *Ack* packet and forwards it back to n_a through n_{m1}.

Second, when a node forwards a *Data* packet, it sets a timer for overhearing subsequent packet forwarded and receiving *Ack* packet from one-hop and two-hop downstream node, respectively. If the node does not overhear subsequent packet forwarded or receive *Ack* packet before its timer expires, because of malicious packet collision or bad channel quality, it suspects the forwarding misbehavior of one-hop downstream node. In the AAA, we propose a simple timeout technique to detect possible packet loss due to malicious packet collision or bad channel quality. We define two timeout periods as a tuple, [T^O, T^A], where T^O and T^A are timeout periods of overhearing packet forwarding and receiving *Ack* packet, respectively. If the packet sender decides not to request the *Ack* packet from two-hop downstream node, its T^A becomes zero. In order to estimate the timeout period, we consider a single-hop average trip time of overhearing packet forwarding (T^O_{avg}) and receiving *Ack* packet (T^A_{avg}), which can be measured by the time from when a node forwards a *Data* packet ($T_{F,Data}$) to when it overhears the packet forwarding ($T_{O,Data}$) and when it receives *Ack* packet ($T_{R,Ack}$), respectively. T^O_{avg} and T^A_{avg} are updated by the low-pass filter with a filter gain constant α. $T^\varphi_{avg} = \alpha \cdot T^\varphi_{avg} + (1-\alpha) \cdot T^\varphi_{k-1}$ and $T^\varphi_{avg} = \frac{\sum_{i=1}^{k-2} T^\varphi_i}{k-2}$. Here, $\varphi \in \{\text{O, A}\}$. T^φ_{k-1} is the measurement from most recently overheard packet forwarding (T^O_{k-1}) and received *Ack* packet (T^R_{k-1}), and they are expressed as, $T^O_{k-1} = T_{O,Data} - T_{F,Data}$ and $T^R_{k-1} = T_{R,Ack} - T_{F,Data}$.

Notations:

- $[T^O, T^A]$, N^{fwd}, N^{ack}, P^{ack}, N^{mis}, N^{unack}, R^{mis}, R^{unack}, ω, τ, δ and c^{mis}: Defined before.
- FS_i, F_j: A set of active forwarding candidate nodes of n_i. A set of one-hop downstream nodes of n_j.
- $pkt[seq, type, m, LP]$: A packet containing a sequence number (seq), packet type ($type$), malicious node id (m), and a list of packet sender of acknowledgment node (LP). Here, $type$ can be *Data*, *Ack*, *Hold*, or *Isolate*.

◇ When a source node, n_s, detects an event:
  Send $pkt[seq, Data, none, LP]$;

◇ When a node, n_j, receives a $pkt[seq, Data, none, LP]$ from n_x:
  if $x \in LP$ /* Packet sender n_x's *id* is in the LP */
    Reply $pkt[seq, Ack, none, none]$ to n_x; /* Reply *Ack* packet */
  $FS_j = \emptyset$;
  for each $n_k \in F_j$
    if n_k is in active state /* n_k is not broadcasting *State* packet */
      $FS_j = FS_j \cup n_k$;
  if $FS_j \neq \emptyset$
    Randomly select a forwarding node, n_f; /* $n_f \in FS_j$ */
    if $P^{ack}_f \leq rand[0,1]$ /* n_j requests *Ack* packet */
      $LP = LP \cup f$; /* Add the *id* of n_f in the LP */
      N^{ack}_f += 1;
      Setup $[T^O, T^A]$;
    else
      Setup $[T^O, none]$;
    Forward $pkt[seq, Data, none, LP]$ to n_f;
    N^{fwd}_f += 1;
  else
    Cache the packet $pkt[seq, Data, none, LP]$;
    Forward $pkt[seq, Hold, none, none]$ to n_x;

◇ When a node, n_i, does not overhear the *Data* packet forwarded or receive *Ack* packet from n_m before T^O or T^A expires, respectively:
  if T^O expires /* n_i does not overhear *Data* packet forwarded */
    N^{mis}_m += 1; $R^{mis}_m = \frac{N^{mis}_m}{N^{fwd}_m}$;
    if $R^{mis}_m \geq \omega$;
      c^{mis}_m++;
      if $c^{mis}_m > \tau$
        Broadcast $pkt[seq, Isolate, m, none]$;
  if T^A expires /* n_i does not receive *Ack* packet */
    N^{unack}_m += 1; $R^{unack}_m = \frac{N^{unack}_m}{N^{ack}_m}$;
    if $R^{unack}_m \geq \omega$;
      c^{mis}_m++; R^{unack}_m += δ;
      if $c^{mis}_m > \tau$
        Broadcast $pkt[seq, Isolate, m, none]$;

Fig. 4. The pseudo code of proposed AAA scheme.

Third, when a malicious node receives the *Data* packet, it may collude with other malicious nodes to drop the packet through packet collision. As shown in Subfig. 3(c), after n_{m1} receives the *Data* packet from n_a, n_{m1} and n_{m2} can communicate through a secret channel to create packet collision. n_{m2} intentionally switches to harvest state and broadcasts *State* packets. n_{m1} and n_{m2} coordinate the forwarding of *Data* packet and the broadcasting of *State* packet simultaneously, resulting in packet collision at n_d. However, this forwarding misbehavior can be detected by the proposed scheme. This is because if the *Data* packet is collided at n_d, n_d will not reply the *Ack* packet back to two-hop upstream node, n_a. Since n_a cannot receive the *Ack* packet before its timer expires, it suspects the forwarding misbehavior of n_{m1}. On the other hand, if n_{m1} simply keeps the *Data* packet without forwarding, n_a will suspect the forwarding misbehavior of n_{m1} since it

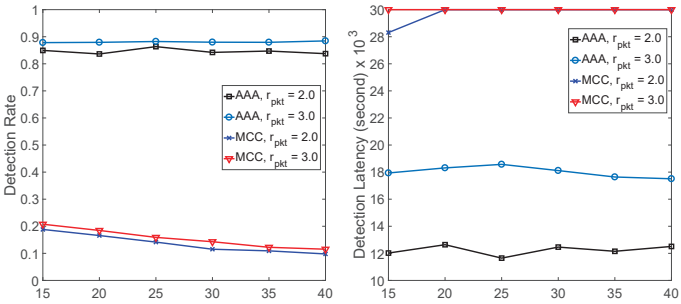
cannot overhear the *Data* packet forwarding before timeout period. n_{m1} may modify the piggybacked *LP* and remove its *id* from the list to disable n_d from replying the *Ack* packet. However, this malicious modification can also be detected since n_a can overhear the forwarded *Data* packet and detect any modification.

Fourth, each node records the number of unoverheard *Data* packets (N^{mis}) and the number of unreceived *Ack* packets (N^{unack}). When a node does not overhear the *Data* packet forwarded or receive the *Ack* packet before the timeout period, it increases the N^{mis} or N^{unack} by one, and then computes the ratio of unoverheard *Data* packet (R^{mis}), $\frac{N^{mis}}{N^{fwd}}$, or the ratio of unreceived *Ack* packet (R^{unack}), $\frac{N^{unack}}{N^{ack}}$. When the updated R^{mis} is equal to or larger than an estimated packet loss ratio, ω, the number of detected forwarding misbehaviors (c_{mis}) is increased by one. When R^{unack} is equal to or larger than ω, c_{mis} and P^{ack} are increased by one and δ, respectively. When the c_{mis} reaches a threshold value τ, the node broadcasts an *Isolate* packet to its all one-hop neighbor nodes to prevent the suspected node from involving any forwarding operation as shown in Subfig. 3(d). Major operations of the proposed AAA scheme are summarized in Fig. 4.

V. Performance Evaluation

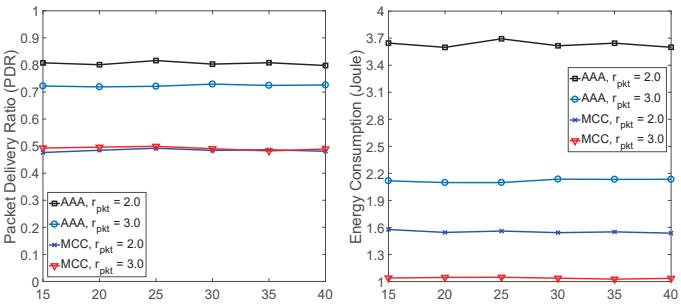
We conduct extensive simulation experiments using OMNeT++ [8] to evaluate the performance of proposed scheme. 250 nodes are uniformly distributed in a 200×200 m^2 rectangular network area. The communication range of each node is 12.3 (m). The radio model simulates CC2420 with a normal data rate of 250 Kbps [19]. A single node generates *Data* traffic with packet injection rate 2.0 and 3.0 pkt/sec and the *Data* packet size is 1 KByte. The inter-arrival time of traffic is assumed to be exponentially distributed. The period of active and harvest states vary between 50 to 80 seconds and 15 to 40 seconds, respectively. The total ten malicious nodes are grouped into five pairs and randomly located in the network. The total simulation time is 30,000 seconds. In this paper, we measure the performance in terms of detection rate, detection latency, packet delivery ratio (PDR), energy consumption, and acknowledgment probability by changing key simulation parameters, including energy harvest time (t_h), packet injection rate (r_{pkt}), and increment weight of acknowledgment probability (δ). For performance comparison, we compare the proposed scheme with the MCC [5].

In Fig. 5, both detection rate and detection latency are measured by changing r_{pkt} and t_h. In Subfig. 5(a), the MCC provides much lower detection rate compared to that of the AAA. This is because adjacent nodes of malicious nodes frequently switch to harvest state and unable to monitor any forwarding operations of malicious nodes. To avoid being detected, the malicious node can advertise the fake number of *Data* packets forwarded to the next hop node during a certain time period. As t_h increases, the detection rate of MCC decreases. This is because each node stays in harvest state for a longer period, and the malicious nodes can collude to



(a) Detection Rate (b) Detection Latency

Fig. 5. The performance of detection rate and detection latency against energy harvest time and packet injection rate.



(a) Packet Delivery Ratio (b) Energy Consumption

Fig. 6. The performance of packet delivery ratio and energy consumption against energy harvest time and packet injection rate.

drop more *Data* packets without being detected. The AAA can achieve much higher and more stable detection rate compared to that of the MCC. Since intermediate nodes are selected to reply the *Ack* packet, the reception of *Data* packets can be confirmed. The detection rate of AAA is not sensitive to the change of t_h. This is because the nodes along the forwarding path rarely switch to harvest state all of sudden. In Subfig. 5(b), the AAA can achieve much lower detection latency compared to that of the MCC. As r_{pkt} decreases, more number of *Data* packet is generated at the source node, and more *Data* packets could be dropped by malicious nodes. Meanwhile, more *Ack* packets are requested and generated to confirm the *Data* packet receptions. Thus, more forwarding misbehaviors can be detected and the malicious nodes can be isolated and removed from the network quickly. Unlike our approach, the MCC shows higher detection latency with r_{pkt} and t_h, because each node only can detect the forwarding misbehavior when overhearing the *Data* packet forwarded in active state. As the node frequently switches to harvest state, more forwarding operations cannot be overheard and thus, it takes much longer time to isolate and remove the malicious nodes from the network.

In Fig. 6, we measure PDR and energy consumption by varying r_{pkt} and t_h. In Subfig. 6(a), the AAA shows higher PDR than that of the MCC, because intermediate nodes actively reply the *Ack* packet back to the upstream nodes and thus, more forwarding misbehaviors can be detected. Finally, the malicious nodes can be quickly isolated and removed from the network, leading to higher PDR. As r_{pkt} increases,

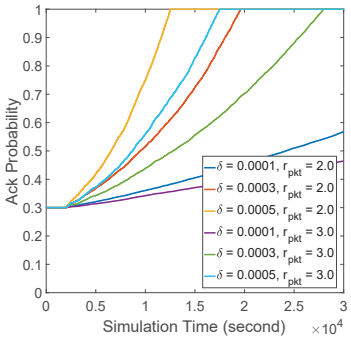


Fig. 7. The performance of acknowledgment probability against simulation time and increment weight.

less number of *Data* packets are generated by the source node, less number of *Data* packets reach to the sink, and the PDR is decreased. The MCC is not sensitive to t_h and r_{pkt}, and the PDR is fluctuating around 50%. This is because the malicious nodes can stay in active state for an extended period and they collude together to create packet collisions with 50% collision rate. In Subfig. 6(b), energy consumption in terms of the number of overheard and received packets [20] is measured. The AAA with different r_{pkt} shows higher energy consumption than that of the MCC because a large number of *Ack* packets is generated and traversed along the forwarding path. Overall energy consumption increases with smaller r_{pkt} = 2.0 pkt/sec because more *Data* packets are generated and more *Ack* packets could be requested and generated along the forwarding path. A lower energy consumption is observed in the MCC. This is because the MCC deploys the implicit monitoring technique, which requires the less number of control packet to detect the forwarding misbehavior and thus, the overall energy consumption is reduced.

We observe the changes of acknowledgment probability P^{ack} with different increment weights (δ = 0.0001, 0.0003, or 0.0005) and r_{pkt} over the simulation period as shown in Fig. 7. If a node cannot receive the requested *Ack* packet before the timeout period, it suspects the forwarding misbehavior of the next hop node and increases P^{ack} by δ. With larger δ, P^{ack} is increased more quickly and more *Ack* packets are requested. Thus, there are more chances to detect the forwarding misbehaviors, leading to a quick isolation of the malicious nodes from the network. For example, the acknowledgment probability reaches to 1.0 at 12,000 seconds with δ = 0.0005. This indicates that any *Data* packet forwarding should be confirmed with *Ack* packet, and any forwarding misbehavior of malicious nodes is suspected and detected. Note that with smaller r_{pkt}, the acknowledgment probability reaches to 1.0 earlier than that of larger r_{pkt}. This is because more *Data* packets are generated with smaller r_{pkt} = 2.0 and more *Ack* packet could be requested, leading to more detected forwarding misbehaviors which results in earlier isolation.

VI. Concluding Remarks

In this paper, we propose a countermeasure to stealthy collision attack in EHNets. A vulnerable scenario motivated by charge-and-spend energy harvesting policy is investigated with a preliminary result. Then an adaptive acknowledgment-based approach, called *AAA*, is proposed to efficiently detect the stealthy collision attack of multiple malicious nodes in EHNets. Extensive simulation results indicate that the proposed countermeasure achieves better performance in terms of detection rate, detection latency and packet delivery ratio compared to the existing implicit monitoring approach, and suggest a new approach to detect the stealthy collision attack in EHNets.

Acknowledgment

This research was supported in part by Startup grant in the Division of Computer Science at Marshall University.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, pp. 1645–1660, 2013.
[2] *A07-034 (Army), Harvesting Energy for Wireless Sensor Networks*, http://www.dodsbir. net/sitis/.
[3] *Killer App: Army Tests Smartphones for Combat*, Wall Street Journal, 6–3–2011.
[4] A. D. Wood and J. A. Stankovic, "Denial of Service in Sensor Networks," *IEEE Computer*, vol. 35, no. 10, pp. 54–62, 2002.
[5] I. Khalil, "MCC: Mitigating Colluding Collision Attacks in Wireless Sensor Networks," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–5.
[6] I. Khalil and S. Bagchi, "Stealthy Attacks in Wireless Ad Hoc Networks: Detection and Countermeasure," *IEEE Trans. on Mobile Computing*, vol. 10, no. 8, pp. 1096–1112, 2011.
[7] C. Pu and S. Lim, "Spy vs. Spy: Camouflage-based Active Detection in Energy Harvesting Motivated Networks," in *Proc. MILCOM*, 2015, pp. 903–908.
[8] A. Varga, *OMNeT++*, 2014, http://www.omnetpp.org/.
[9] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proc. ACM MOBICOM*, 2000, pp. 255–265.
[10] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs," *IEEE Trans. on Mobile Computing*, vol. 6, no. 5, pp. 536–550, 2007.
[11] C. Pu and S. Lim, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation," *IEEE Systems Journal*, 2016.
[12] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Adaptive and Channel-Aware Detection of Selective Forwarding Attacks in Wireless Sensor Networks," *IEEE Trans. on Wireless Commun.*, vol. 15, no. 5, pp. 3718–3731, 2016.
[13] S. Lim and H. Lauren, "Hop-by-Hop Cooperative Detection of Selective Forwarding Attacks in Energy Harvesting Wireless Sensor Networks," in *Proc. ICNC*, 2015, pp. 315–319.
[14] F. K. Shaikh and S. Zeadally, "Energy Harvesting in Wireless Sensor Networks: A Comprehensive Review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, 2016.
[15] S. Lim, J. Kimn, and H. Kim, "Analysis of Energy Harvesting for Vibration-Motivated Wireless Sensor Networks," in *Proc. ICWN*, 2010, pp. 391–397.
[16] C. Pu, T. Gade, S. Lim, M. Min, and W. Wang, "Light-Weight Forwarding Protocols in Energy Harvesting Wireless Sensor Networks," in *Proc. MILCOM*, 2014, pp. 1053–1059.
[17] W. Stallings, *Cryptography and Network Security - Principles and Practices, 6th Edition*. Prentice Hall, 2013.
[18] R. Chen and Y. Wang, "Reliability Analysis of Wireless Sensor Networks with Distributed Code Attestation," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1640–1643, 2012.
[19] A. Boulis, *Castalia*, 2014, http://castalia.forge.nicta.com.au.
[20] X. Tang and J. Xu, "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," in *Proc. INFOCOM*, 2006, pp. 1–12.