
Jun-Won Ho, Matthew Wright, and Sajal K. Das
Department of Computer Science and Engineering
The University of Texas at Arlington
jun-won.ho@mavs.uta.edu, {mwright, das}@uta.edu

Abstract—Due to the unattended nature of wireless sensor networks, an adversary can capture and compromise sensor nodes, make replicas of them, and then mount a variety of attacks with these replicas. These replica-node attacks are dangerous because they allow the attacker to leverage the compromise of a few nodes to exert control over much of the network. Several replica node detection schemes have been proposed in the literature to defend against such attacks in static sensor networks. However, these schemes rely on fixed sensor locations and hence do not work in mobile sensor networks, where sensors are expected to move. In this work, we propose a fast and effective mobile replica node detection scheme using the Sequential Probability Ratio Test. To the best of our knowledge, this is the first work to tackle the problem of replica node attacks in mobile sensor networks. We show analytically and through simulation experiments that our scheme detects mobile replicas in an efficient and robust manner at the cost of reasonable overheads.

Index Terms—Replica detection, sequential analysis, mobile sensor networks, security.

I. INTRODUCTION

Advances in robotics have made it possible to develop a variety of new architectures for autonomous wireless networks of sensors. Mobile nodes, essentially small robots with sensing, wireless communications, and movement capabilities, are useful for tasks such as static sensor deployment, adaptive sampling, network repair, and event detection [4]. These advanced sensor network architectures could be used for a variety of applications including intruder detection, border monitoring, and military patrols. In potentially hostile environments, the security of unattended mobile nodes is extremely critical. The attacker may be able to capture and compromise mobile nodes, and then use them to inject fake data, disrupt network operations, and eavesdrop on network communications.

In this scenario, a particularly dangerous attack is the replica node attack [11], in which the adversary takes the secret keying materials from a compromised node, generates a large number of attacker-controlled replicas that share the compromised node’s keying materials and ID, and then spreads these replicas throughout the network. With a single captured node, the adversary can create as many replica nodes as he has the hardware to generate. Note that replica nodes need not be identical robots; a group of static nodes can mimic the movement of a robot and other mobile nodes or even humans with handheld devices could be used. The only requirement is that they have the software and keying material to communicate in the network, all of which can be obtained from the captured node.

The time and effort needed to inject these replica nodes into the network should be much less than the effort to capture and compromise the equivalent number of original nodes. The replica nodes are controlled by the adversary, but have keying materials that allow them to seem like authorized participants in the network. Protocols for secure sensor network communication would allow replica nodes to create pairwise shared keys with other nodes and the base station, thereby enabling the nodes to encrypt, decrypt, and authenticate all of their communications as if they were the original captured node.

The adversary can then leverage this insider position in many ways. For example, he can simply monitor a significant fraction of the network traffic that would pass through these nodes. Alternately, he could jam legitimate signals from benign nodes or inject falsified data to corrupt the sensors’ monitoring operation. A more aggressive attacker could undermine common network protocols, including cluster formation, localization, and data aggregation, thereby causing continual disruption to network operations. Through these methods, an adversary with a large number of replica nodes can easily defeat the mission of the deployed network.

A straightforward solution to stop replica node attacks is to prevent the adversary from extracting secret key materials from mobile nodes by equipping them with tamper-resistant hardware. We might expect such measures to be implemented in mobile nodes with security-critical missions. However, although tamper-resistant hardware can make it significantly harder and more time-consuming to extract keying materials from captured nodes, it may still be possible to bypass tamper resistance for a small number of nodes given enough time and attacker expertise. Since the adversary can generate many replicas from a single captured node, this means that replica attacks are even more dangerous when compared with the possibility of compromising many nodes. We thus believe that it is very important to develop software-based countermeasures to defend mobile sensor networks against replica node attacks.

Several software-based replica node detection schemes have been proposed for static sensor networks [3], [11], [17]. The primary method used by these schemes is to have nodes report
location claims that identify their positions and for other nodes to attempt to detect conflicting reports that signal one node in multiple locations. However, since this approach requires fixed node locations, it cannot be used when nodes are expected to move. Thus, our challenge is to design an effective, fast, and robust replica detection scheme specifically for mobile sensor networks.

In this paper, we propose a novel mobile replica detection scheme based on the Sequential Probability Ratio Test (SPRT) [15]. We use the fact that an uncompromised mobile node should never move at speeds in excess of the system-configured maximum speed. As a result, a benign mobile sensor node’s measured speed will nearly always be less than the system-configured maximum speed as long as we employ a speed measurement system with a low error rate. On the other hand, replica nodes are in two or more places at the same time. This makes it appear as if the replicated node is moving much faster than any of the benign nodes, and thus the replica nodes’ measured speeds will often be over the system-configured maximum speed. Accordingly, if we observe that a mobile node’s measured speed is over the system-configured maximum speed, it is then highly likely that at least two nodes with the same identity are present in the network.

However, if the system decides that a node has been replicated based on a single observation of a node moving faster than it should, we might get many false positives because of errors in speed measurement. Raising the speed threshold or other simple ways of compensating can lead to high false negative rates. To minimize these false positives and false negatives, we apply the SPRT, a hypothesis testing method that can make decisions quickly and accurately. We perform the SPRT on every mobile node using a null hypothesis that the mobile node has not been replicated and an alternate hypothesis that it has been replicated. In using the SPRT, the occurrence of a speed that is less than or exceeds the system-configured maximum speed will lead to acceptance of the null or alternate hypotheses, respectively. Once the alternate hypothesis is accepted, the replica nodes will be revoked from the network.

We validate the effectiveness, efficiency, and robustness of our scheme through analysis and simulation experiments. Specifically, we find that the main attack against the SPRT-based scheme is when replica nodes fail to provide signed location and time information for speed measurement. To overcome this attack, we employ a quarantine defense technique to block the non-compliant nodes. We then study this technique in two ways. First, we show through quarantine analysis that the amount of time, during a given time slot, that the replicas can impact the network is very limited. Second, we provide a detailed game-theoretic analysis that shows the limits of any attacker strategy over any number of time slots. Specifically, we formulate a two-player game to model the interaction between the attacker and the defender, derive the optimal attack and defense strategies, and show that the attacker’s gain is greatly limited when the attacker and the defender follow their respective optimal strategies. We provide analyses of the number of speed measurements needed to make replica detection decisions, which we show is quite low, and the amount of overhead incurred by running the protocol.

We also evaluate the performance of our scheme via simulation study using ns-2 simulator. In particular, we consider two types of replicas for performance evaluation: mobile and static. In case of mobile replicas, we investigate how replica mobility affects the detection capability of our scheme. In case of static (immobile) replicas, the attacker keeps his replica nodes close together and immobile to lessen the chance of speed-based detection. An exploration of the static replica case is useful since this case represents the worst case for detection, and thus we can see how our scheme works in the worst case. The simulation results of both cases show that this scheme very quickly detects mobile replicas with low false positive and negative rates. A preliminary version of this work appeared in [5].

The rest of paper is organized as follows. Section II describes the problem statement, network assumptions, and adversary models for our scheme. Section III presents the proposed mobile replica detection scheme using the SPRT along with security and performance analyses. Section IV presents the results of simulations we conducted to evaluate the proposed scheme. Section V presents the related work. Finally, Section VI concludes the paper.

II. PROBLEM DEFINITION

In this section, we first state the problem and the network assumptions for our proposed scheme and then describe the attacker models we use to evaluate our approach.

A. Problem Statement

We define a mobile replica node \( u' \) as a node having the same ID and secret keying materials as a mobile node \( u \). An adversary creates replica node \( u' \) as follows: He first compromises node \( u \) and extracts all secret keying materials from it. Then he prepares a new node \( u' \), sets the ID of \( u' \) to the same as \( u \), and loads \( u \)'s secret keying materials into \( u' \). There may be multiple replicas of \( u \), e.g. \( u_1' \), \( u_2' \), . . . , and there may be multiple compromised and replicated nodes. Our goal is to detect the fact that both \( u \) and \( u' \) (or \( u_1' \), \( u_2' \), . . . ) operate as separate entities with the same identity and keys.

B. Network Assumptions

We consider a two-dimensional mobile sensor network where sensor nodes freely roam throughout the network. We assume that every mobile sensor node’s movement is physically limited by the system-configured maximum speed, \( V_{max} \). We also assume that all direct communication links between sensor nodes are bidirectional. This communication model is common in the current generation of sensor networks. We assume that every mobile sensor node is capable of obtaining its location information and also verifying the locations of its neighboring nodes. This can be implemented by employing secure localization methods [2], [7]. We assume that the clocks of all nodes are loosely synchronized. This can be achieved with the help of secure time synchronization protocols [12], [13]. We also assume that the nodes in the mobile sensor
network communicate with a base station. The base station may be static or mobile, although we focus on a static base station for our simulations, as long as the nodes have a way to communicate reliably to the base station on a regular basis.

C. Attacker Models

We assume that an adversary may compromise and fully control a subset of the sensor nodes, enabling him to mount various kinds of attacks. For instance, he can inject false data packets into the network and disrupt local control protocols such as localization, time synchronization, and route discovery process. Furthermore, he can launch denial of service attacks by jamming the signals from benign nodes. However, we place some limits on the ability of the adversary to compromise nodes. We note that if the adversary can compromise a major fraction nodes of the network, he will not need nor benefit much from the deployment of replicas.

To amplify his effectiveness, the adversary can also launch a replica node attack, which is the subject of our investigation. We assume that the adversary can produce many replica nodes and that they will be accepted as a legitimate part of the network. We also assume that the attacker attempts to employ as many replicas of one or more compromised sensor nodes in the network as will be effective for his attacks. The attacker can allow his replica nodes to randomly move or he could move his replica nodes in different patterns in an attempt to frustrate our proposed scheme. We discuss this possibility in Section III-B.

We also assume that the base station is a trusted entity. This is a reasonable assumption in mobile sensor networks, because the network operator collects all sensor data and can typically control the nodes’ operation through the base station. Thus, the basic mission of the sensor network is already completely undermined if the base station is compromised.

III. MOBILE REPLICA DETECTION USING SEQUENTIAL PROBABILITY RATIO TEST

This section presents the details of our technique to detect replica node attacks in mobile sensor networks.

In static sensor networks, a sensor node is regarded as being replicated if it is placed in more than one location. If nodes are moving around in network, however, this technique does not work, because a benign mobile node would be treated as a replica due to its continuous change in location. Hence, we must use some other technique to detect replica nodes in mobile sensor networks. Fortunately, mobility provides us with a clue to help resolve the mobile replica detection problem. Specifically, a benign mobile sensor node should never move faster than the system-configured maximum speed, $V_{\text{max}}$. As a result, a benign mobile sensor node’s measured speed will appear to be at most $V_{\text{max}}$ as long as we employ a speed measurement system with a low rate of error. On the other hand, replica nodes will appear to move much faster than benign nodes and thus their measured speeds will likely be over $V_{\text{max}}$ because they need to be at two (or more) different places at once. Accordingly, if the mobile node’s measured speed exceeds $V_{\text{max}}$, it is then highly likely that at least two nodes with the same identity are present in the network.

We propose a mobile replica detection scheme by leveraging this intuition. Our scheme is based on the Sequential Probability Ratio Test (SPRT) [15] which is a statistical decision process. The SPRT can be thought of as one-dimensional random walk with the lower and upper limits [8]. Before the random walk starts, null and alternate hypotheses are defined in such a way that the null hypothesis is associated with the lower limit while the alternate one is associated with the upper limit. A random walk starts from a point between two limits and moves toward the lower or upper limit in accordance with each observation. If the walk reaches (or exceeds) the lower or upper limit, it terminates and the null or alternate hypothesis is selected, respectively. We believe that the SPRT is well suited for tackling the mobile replica detection problem since we can construct a random walk with two limits in such a way that each walk is determined by the observed speed of a mobile node. The lower and upper limits can be configured to be associated with speeds less than and in excess of $V_{\text{max}}$, respectively.

We apply the SPRT to the mobile replica detection problem as follows. Each time a mobile sensor node moves to a new location, each of its neighbors asks for a signed claim containing its location and time information and decides probabilistically whether to forward the received claim to the base station. The base station computes the speed from every two consecutive claims of a mobile node and performs the SPRT by considering speed as an observed sample. Each time the mobile node’s speed exceeds (resp. remains below) $V_{\text{max}}$, it will expedite the random walk to hit or cross the upper (resp. lower) limit and thus lead to the base station accepting the alternate (resp. null) hypothesis that the mobile node has been (resp. not been) replicated. Once the base station decides that a mobile node has been replicated, it revokes the replica nodes from the network.

Let us first describe the detection scheme and then analyze its security and performance.

A. Protocol Description

Before deployment, every sensor node gets secret keying materials for generating digital signatures. We will use an identity-based public key scheme. It has been demonstrated that public key operations can be efficiently implemented in static sensor devices [9], [16]. Moreover, most replica detection schemes in static sensor networks [3], [11] employ identity-based public key signatures. Mobile sensor devices are generally more powerful than static ones in terms of battery power, due to the fact that the mobile sensor node consumes a lot of energy to move. Additionally, the energy consumption due to movement is known to be substantially larger than that for public key operations. For example, the power consumption for the movement of a mobile sensor device has been measured at 720 mW [4]. The energy consumption for computing and verifying a public key signature have been measured at between 2.9 mW to 48 mW and between 3.5 mW to 58.5 mW, respectively, in accordance with existing sensor
1) Claim Generation and Forwarding: Each time a mobile sensor node \( u \) moves to a new location, it first discovers its location \( L_u \) and then discovers its set of neighboring nodes, \( N(u) \). Every neighboring node \( v \in N(u) \) asks node \( u \) for an authenticated location claim by sending its current time \( T \) to node \( u \). Upon receiving \( T \), node \( u \) checks whether \( T \) is valid or not. If \( |T' - T| > \delta + \epsilon \), where \( T' \) is the claim receipt time at \( u \), \( \delta \) is the estimated transmission delay of claim, and \( \epsilon \) is a maximum error in time synchronization, then node \( u \) will ignore the request. Otherwise, \( u \) generates location claim \( C_u = \{u||L_u||T||\text{Sig}_u\} \) and sends it to \( v \), where \( \text{Sig}_u \) is the signature over the tuple \( (u, L_u, T) \) generated using node \( u \)’s private key. If \( u \) denies the claim requests, or if its claim contains invalid time information or fails to authenticate, then \( u \) will be removed from \( N(v) \). Also, if \( u \) claims a location \( L_u \) such that the distance between \( L_u \) and \( L_v \) is larger than the assumed signal range of \( v \), then it will be removed from \( N(v) \).

Once the above filtering process is passed, each neighbor \( v \) of node \( u \) forwards \( u \)’s claim to the base station with probability \( p \).

Regarding errors in the measurement of time and location, we can consider both random and systematic errors. Since speed is measured based on location and time, the errors can come from either measurement. We note that the time of each claim is measured and verified by the requesting node, thus we believe that a public key signature scheme can be practical for mobile sensor networks. Our proposed protocol proceeds in two phases.

2) Detection and Revocation: Upon receiving a location claim from node \( u \), the base station verifies the authenticity of the claim with the public key of \( u \) and discards the claim if it is not authentic. We note the authentic claims from node \( u \) by \( C_u \), \( C_u^2 \), \ldots . The base station extracts location information \( L_i^u \) and time information \( T_i \) from claim \( C_u^i \). Let \( d_i \) denote the Euclidean distance from location \( L_i^u \) at time \( T_i \) to \( L_{i+1}^u \) at \( T_{i+1} \). Let \( o_i \) denote the measured speed at time \( T_{i+1} \), where \( i = 1, 2, \ldots \). In other words, \( o_i \) is defined as:

\[
o_i = \frac{d_i}{|T_{i+1} - T_i|}
\]

Let \( S_i \) denote a Bernoulli random variable defined as:

\[
S_i = \begin{cases} 
0 & \text{if } o_i \leq V_{\max} \\
1 & \text{if } o_i > V_{\max}
\end{cases}
\]

Then the success probability \( \lambda \) of the Bernoulli distribution is defined as:

\[
Pr(S_i = 1) = 1 - Pr(S_i = 0) = \lambda
\]

If \( \lambda \) is less than or equal to a preset threshold \( \lambda' \), it is likely that node \( u \) has not been replicated. On the other hand, if \( \lambda > \lambda' \), it is likely that node \( u \) has been replicated. The problem of deciding whether \( u \) has been replicated or not can be formulated as a hypothesis testing problem with null and alternate hypotheses of \( \lambda \leq \lambda' \) and \( \lambda > \lambda' \), respectively. In this problem, we need to devise an appropriate sampling strategy in order to prevent hypothesis testing from making the wrong decision. In particular, we should specify the maximum chance errors that we want to tolerate for a good sampling strategy. To do this, we reformulate the above hypothesis testing problem as one with null and alternate hypotheses of \( \lambda \leq \lambda_0 \) and \( \lambda \geq \lambda_1 \), respectively, such that \( \lambda_0 < \lambda_1 \).

To understand the basis of this sampling plan, we present how the SPRT is performed to make a decision about node \( u \) from the \( n \) observed samples, where a measured speed of \( u \) is treated as a sample. We first define the null hypothesis \( H_0 \) and the alternate one \( H_1 \) as follows: \( H_0 \) is the hypothesis that node \( u \) has not been replicated and \( H_1 \) is the hypothesis that \( u \) has been replicated. We then define \( L_n \) as the log-probability ratio on \( n \) samples, given as:

\[
L_n = \ln \frac{Pr(S_1, \ldots, S_n|H_1)}{Pr(S_1, \ldots, S_n|H_0)}
\]

We assume that each speed measurement for a given node is independent of the other speed measurements. Thus, we assume that \( S_i \) is independent and identically distributed (i.i.d.). Then, \( L_n \) can be rewritten as:

\[
L_n = \ln \frac{\prod_{i=1}^{n} Pr(S_i|H_1)}{\prod_{i=1}^{n} Pr(S_i|H_0)} = \sum_{i=1}^{n} \ln \frac{Pr(S_i|H_1)}{Pr(S_i|H_0)}
\]

Let \( \omega_n \) denote the number of times that \( S_i = 1 \) in the \( n \) samples. Thus we have

\[
L_n = \omega_n \ln \frac{\lambda_1}{\lambda_0} + (n - \omega_n) \ln \frac{1 - \lambda_1}{1 - \lambda_0}
\]

Where:

\[
\lambda_0 = Pr(S_1 = 1|H_0), \lambda_1 = Pr(S_1 = 1|H_1).
\]

The rationale behind the configuration of \( \lambda_0 \) and \( \lambda_1 \) is as follows. On the one hand, \( \lambda_0 \) should be configured in accordance with the likelihood of the occurrence that a benign node’s measured speed exceeds \( V_{\max} \) due to time synchronization and localization errors. On the other hand, \( \lambda_1 \) should be configured to consider the likelihood of the occurrence that replica nodes’
measured speeds exceed \(V_{\text{max}}\). Since the former likelihood is lower than the latter one, \(\lambda_0\) should be set lower than \(\lambda_1\).

On the basis of the log-probability ratio \(L_n\), the SPRT for \(H_0\) against \(H_1\) is given as follows:

- \(L_n \leq \ln \frac{\lambda_0}{\lambda_1} \): accept \(H_0\) and terminate the test.
- \(L_n \geq \ln \frac{1-\beta'}{\alpha} \): accept \(H_1\) and terminate the test.
- \(\ln \frac{1-\beta'}{\alpha} < L_n < \ln \frac{1-\beta'}{\alpha} \): continue the test process with another observation.

We can rewrite the SPRT as follows:

- \(\omega_n \leq \tau_0(n) \): accept \(H_0\) and terminate the test.
- \(\omega_n \geq \tau_1(n) \): accept \(H_1\) and terminate the test.
- \(\tau_0(n) < \omega_n < \tau_1(n) \): continue the test process with another observation.

Where:

\[
\tau_0(n) = \frac{\ln \frac{\beta'}{1-\beta'} + n \ln \frac{\lambda_0}{\lambda_1}}{\ln \frac{\lambda_0}{\lambda_1} - \ln \frac{\lambda_0}{\lambda_1}}, \quad \tau_1(n) = \frac{\ln \frac{1-\beta'}{\alpha} + n \ln \frac{1-\lambda_0}{\lambda_1}}{\ln \frac{\lambda_0}{\lambda_1} - \ln \frac{\lambda_0}{\lambda_1}}
\]

If a mobile node \(u\) is judged as benign, the base station restarts the SPRT with newly arrived claims from \(u\). If, however, \(u\) is determined to be replicated, the base station terminates the SPRT on \(u\) and revokes all nodes with identity \(u\) from the network.

### B. Security Analysis

In this section, we will first describe the detection accuracy of our proposed scheme and then present attack scenarios to break this scheme and a defense strategy we propose to limit these attacks. Finally, we will show that the attacker’s gain is substantially limited by the defense strategy.

1) Detection Accuracy: In the SPRT, \(\alpha\) and \(\beta\) are defined as the error probability that the SPRT accepts \(H_1\) (resp. \(H_0\)) when \(H_0\) (resp. \(H_1\)) is true. Since \(H_0\) is the hypothesis that a node \(u\) has not been replicated, \(\alpha\) and \(\beta\) are the false positive and false negative probabilities of the SPRT, respectively. According to Wald’s theory [15], the upper bounds of \(\alpha\) and \(\beta\) are calculated as \(\alpha \leq \frac{\alpha'}{1-\beta'}\) and \(\beta \leq \frac{\beta}{1-\alpha}\), respectively. Furthermore, it has been shown [15] that the sum of the false positive and negative probabilities of the SPRT is limited by the sum of user-configured false positive and negative probabilities. Namely, the inequality \(\alpha + \beta \leq \alpha' + \beta'\) holds. Since \(\beta\) is the false negative probability, \((1-\beta)\) is the replica detection probability. Accordingly, the lower bound on the replica detection probability is \((1-\beta) \geq \frac{1-\alpha' - \beta'}{1-\alpha' - \beta'}\). From the above inequalities, we observe that low user-configured false positive and negative probabilities will lead to a low false negative probability for the sequential test process. Hence, it will result in high detection rates. For instance, if the user configures both \(\alpha'\) and \(\beta'\) to 0.01, then the replica detection is guaranteed with probability 0.99.

2) Limitations of Replica Node Attacks: Let us now discuss in which the attacker could attempt to evade our detection scheme and defensive countermeasures that we can employ.

First, a malicious node \(u\) may attempt to forge a claim, either by sending a claim with incorrect data or by sending a claim with a bad signature. However, all of \(u\)‘s neighbors will check the validity of \(u\)’s identity, reported location, reported time, and the signature over these values using node \(u\)’s public key. Alternatively, node \(u\) can simply ignore the claim requests. In our scheme, if \(u\)’s benign neighbor does not receive a claim despite sending a claim request, it will remove \(u\) from its neighboring set and will not communicate with \(u\).

We note that if one of \(u\)’s neighbors is malicious, the malicious node can serve as \(u\)’s neighbor for forwarding packets. However, there is little benefit to the attacker of having a replica node in the same area as another compromised node. The compromised node can just as easily report fake data, participate in local control protocols, and eavesdrop on messages sent through it. Furthermore, if the attacker needs one compromised node to accompany each replica node in the network, there will be a very high cost for replica node attacks.

Similarly, an attacker will not gain much benefit from having multiple replicas of a single node form a group that always moves together and stays close enough so that all replicas can claim the same location. This is because these nodes would essentially have the same set of neighbors. Consider a compromised node \(u\) and its replica \(u'\) communicating with neighboring node \(v\). From \(v\)’s perspective, there is no difference between the two replicas, and \(v\) treats all messages as coming from a single node. The two nodes thus can not do anything that a single compromised node \(u\) could not do by itself. If the replicas can claim the same location while reaching a slightly larger set of neighbors, then the attacker can gain a small amount of additional influence through the replica attack, but no more than it could gain with a better antenna and more signal power.

An interesting variant of this attack, however, is to keep replicas close to each other so that the perceived velocity between their location claims is less than \(V_{\text{max}}\). To do this, an attacker coordinates a set of replicas to respond with correct claims only to those claim requests that make it appear as a single node never moving faster than \(V_{\text{max}}\). The attacker can have some replicas grouped closely together for this purpose; replicas that are further away must ignore claim requests or respond with false claims to avoid detection. To illustrate this attacker’s strategy, let us consider a simple attack scenario in which a compromised node \(u\) and its replica \(u'\) are fixed to some locations in such a way that the distance between these two nodes is set to \(d\). We assume that nodes \(u\) and \(u'\) initiate the neighbor discovery process at time \(T_0\) and \(T_0 + \frac{d}{V_{\text{max}}}\), respectively. Moreover, suppose that node \(u\) receives a claim request from neighbor \(v\) at time \(T_0 + \frac{d}{V_{\text{max}}} + \xi\) and node \(u'\) receives a request from neighbor \(w\) at time \(T_0 + \frac{d}{V_{\text{max}}} + \xi\). Nodes \(u\) and \(u'\) send claims to \(v\) and \(w\) and ignore all incoming claim requests from other neighbors or give them false claims. Even though either \(v\) or \(w\) may move to a new location, the attacker can control nodes \(u\) and \(u'\) to accept claim requests from newly designated neighbors in such a way that the claim receipt time of \(u\) remains \(\frac{d}{V_{\text{max}}}\) time ahead of that of \(u'\). In this way, the attacker can successfully deceive the base station to believe that \(u\) moves back and forth with speed \(V_{\text{max}}\). This attack scenario can be generalized to the case of a set of replicas and to allow for movement.
Since the replicas do not provide valid claims that would make the observed speed exceed $V_{\text{max}}$, they can trick the base station into accepting $H_0$, the hypothesis that they are not replicas. To stop this attack, we propose to have the base station check whether each node responds with correct claims to all incoming claim requests.

Specifically, each time a malicious node $u$ ignores a claim request from a benign neighbor node $v$ or responds with false claims, $v$ generates a denial of claim request notification message, $DCN = \{v||u||MAC_K, [v]|u]\}$ and sends it to the base station, where $MAC$ is a message authentication code calculated using $K_v$, a the shared secret key between $v$ and the base station. Upon receiving the $DCN$ message from $v$, the base station first checks the authenticity of the $DCN$ and rejects it if it is invalid. Assume that the entire time domain is divided into time slots. The base station maintains a $DCN$ counter for each node such that it initializes each counter to 0 and then resets it to 0 at the beginning of each time slot. Each time the base station receives a $DCN$ message on $u$ from $v$, it increases the $DCN$ counter for $u$. If the $DCN$ counter for $u$ exceeds a predefined threshold $\rho$ during a time slot, it is highly likely that $u$ has discarded a substantial fraction of claim requests during the time slot and is likely to be a replica node attempting to evade detection.

In this case, the base station will temporarily quarantine $u$ from the network by disregarding all messages from $u$ and broadcasting the quarantine information to all nodes. Upon receiving this quarantine message, all nodes will stop communicating with $u$ except for exchanging claim request and response messages. If the $DCN$ counter for $u$ does not exceed threshold $\rho$ during the quarantine period, the base station will release the quarantine that it imposed on $u$ after the expiry of the quarantine period and broadcast the release information. Otherwise, it will extend the quarantine period by one time slot. The quarantine period needs to long enough to ensure that the replica nodes would be quarantined for longer periods than they would be able to participate freely in the network. This principle will also help prevent frequent oscillation between quarantine and non-quarantine states. Since the base station determines in each time slot whether to impose quarantine on a node, it can satisfy the above principle by setting the quarantine period to be multiple time slots.

To trick the base station into putting benign nodes into the quarantine, the attacker could send many fake $DCN$ messages. Specifically, if the base station receives more than $\rho$ fake $DCN$ messages on the benign node $v$, then $v$ will be quarantined even though it responds correctly to all incoming claim requests. To discourage this type of attack, we restrict each node from sending more than one $DCN$ per time slot. If the base station receives more than one $DCN$ from a node during a time slot, it will accept only one $DCN$ from the node and discard the others. Hence, the attacker needs more than $\rho$ compromised nodes per time slot to force a benign node to be quarantined, thus suppressing him from mounting a fake $DCN$ attack. From this analysis, we see that $\rho$ needs to be configured in accordance with the number of compromised nodes in the network. From the perspective that the main benefit of replica node attacks is to substantially reduce the time and efforts required for wide-spread node compromise, the attacker will be interested in having only a few compromised nodes when he employs replica node attacks. Therefore, it will be reasonable for $\rho$ to be set to a small value in practice.

3) Short-Term Quantitative Analysis of Quarantine Defense Strategy: We now quantitatively determine a limit on the amount of time for which a set of replicas can avoid detection and quarantine when they follow a strategy of responding only to selected claims. Our underlying argument is that the replica nodes must ignore a minimum number of claim requests to avoid detection, but we will configure the quarantine system to react and stop the replica node attacks when many claims are ignored.

Suppose that $r$ replicas of a compromised node $u$ are fixed to some locations. We model the arrival of claim requests to each replica as a homogeneous poisson process. We use a poisson process due to the following reasons: First, we assume that mobile nodes’ movements in disjoint intervals are independent from each other and thus the number of times that mobile nodes meet to replicas in disjoint intervals are accordingly independent from each other. Second, the probability distribution of the number of claim requests received by replicas in a time interval should be modeled to only depend on the length of the interval. This is reasonable in the sense that the number of claim requests received by replicas in a time interval varies in accordance with the length of the interval.

Note that the sum of multiple homogeneous poisson processes is also a homogeneous poisson process with the rates summed together. Thus, we model the claim request arrival process of the $r$ replicas as the homogeneous poisson process with rate parameter $\theta$. Let $X_i$, $i = 1, 2, \ldots$ be independent, identically distributed (i.i.d.) exponential random variables with parameter $\theta$.

Let $\Delta T$ denote the duration of a time slot. Suppose that claim requests arrive at $r$ replicas from the beginning to the end of a time slot. Let $o_i$ be the speed measured according to the $i$th and the $(i+1)$th claim requests during $\Delta T$ time. Thus, $o_i$ is defined as $\frac{d_i}{\Delta T}$, where $d_i$ is the Euclidean distance between a pair of replicas that receive the $i$th and the $(i+1)$th claim requests. Let $p'(d_i)$ be the probability that $o_i$ exceeds $V_{\text{max}}$ during $\Delta T$ time, where $\frac{d_i}{V_{\text{max}}} \leq \Delta T$. Clearly, $p'(d_i)$ is equivalent to the probability $\Pr(X_i < \frac{d_i}{V_{\text{max}}} \mid X_i \leq \Delta T)$. Since $X_i$ follows an exponential distribution with parameter $\theta$, we compute the probability $p'(d_i)$ with the aid of the improper integral:

$$p'(d_i) = \Pr \left( X_i < \frac{d_i}{V_{\text{max}}} \mid X_i \leq \Delta T \right)$$

$$= \lim_{\Delta T \to \infty} \frac{1 - e^{-\theta \Delta T}}{1 - e^{-\theta \frac{d_i}{V_{\text{max}}}}}$$

We define $Y_i$ as a Bernoulli random variable with success
probability \( p'(d_i) \) such that:

\[
Y_i = \begin{cases} 
0 & \text{if } X_i \geq \frac{d_i}{V_{max}} \text{ given } X_i \leq \Delta T \\
1 & \text{if } X_i < \frac{d_i}{V_{max}} \text{ given } X_i \leq \Delta T 
\end{cases}
\]

Note that \( \mu = \theta \times \Delta T \) is the expected number of events that occur during \( \Delta T \) time in the homogeneous poisson process with rate \( \theta \). Accordingly, the expected number of claim requests during \( \Delta T \) time is \( \mu \). By considering the claim forwarding probability \( p \), the expected number of legitimate claims forwarded to the base station during \( \Delta T \) time is at most \( \rho \mu \), corresponding to \( \rho \mu - 1 \) samples. Hence, \( \sum_{i=1}^{\rho \mu - 1} p'(d_i) \) is the expected number of times that \( Y_i = 1 \) in \( \rho \mu - 1 \) samples.

Let \( d_{min} \) be the shortest distance between a pair of replicas. Since \( p'(d_{min}) \leq p'(d_i) \), the sum \( \sum_{i=1}^{\rho \mu - 1} p'(d_i) \) should be no less than \( p'(d_{min}) \times (\rho \mu - 1) \). Therefore, the expected number of samples that cause the measured speed to exceed \( V_{max} \) during \( \Delta T \) time is bounded from below by \( p'(d_{min}) \times (\rho \mu - 1) \).

As a consequence, if \( p'(d_{min}) \times (\rho \mu - 1) > [\tau_1(\rho \mu - 1)] - 1 \), the replicas should ignore at least \( \Delta R \) claim requests during \( \Delta T \) time in order to prevent them from being detected, where \( \Delta R = p'(d_{min}) \times (\rho \mu - 1) - [\tau_1(\rho \mu - 1)] + 1 \). However, if they ignore all \( \Delta R \) requests, they will be quarantined as long as \( \Delta R > \rho \). In this sense, the replicas are stuck between getting detected and quarantined. Thus, we should set \( \rho \) to be less than \( \Delta R \) to ensure that we either detect or quarantine replicas.

Let us now investigate how to configure \( \Delta T \) to keep \( \Delta R > \rho \) under different settings of distance \( d_{min} \) and claim request arrival rate \( \theta \). For this purpose, we use the following fixed configuration: \( V_{max} = 20 \text{ m/s}, p = 0.05, \lambda_0 = 0.1, \lambda_1 = 0.95, \alpha' = 0.01, \beta' = 0.01 \). We consider three different cases as shown in Figure 1. In all three cases, \( \Delta R \) mainly increases with \( \Delta T \). This implies that \( \Delta T \) needs to be configured in proportion to \( \rho \) to ensure that \( \Delta R > \rho \). For instance, if we set \( \rho = 4 \), \( \Delta T \) should be more than 10, 5, and 4 seconds in Cases I, II, and III, respectively, so as to ensure that \( \Delta R > \rho \). This means that the replicas can avoid quarantine during less than \( \Delta T \) time, which is just a few seconds when \( \rho = 4 \) in our example scenarios.

Finally, the attacker’s only option to avoid detection and quarantine is to move the replicas to an entirely new location before the arrivals of the claim requests that force replicas to be detected or quarantined. However, this will allow the replicas to evade detection and quarantine only during less than \( \Delta T \) time and thus will greatly limit the attacker’s ability to control parts of the network for any length of time.

4) Long-Term Game-Theoretic Analysis of Quarantine Defense Strategy: Through the above analysis, we showed that the quarantine defense strategy substantially restricts the attacker’s gains from employing a selective claim request responding strategy when the duration of a single time slot is reasonably configured. However, the above analysis does not fully reflect the interactions between attacker and defender, since it focuses on a single time slot. To analyze how the quarantine strategy provides resilience against the selective claim request responding strategy for a long period of time, we develop a game theoretic model of claim response and quarantine defense. This model is useful to understand what the optimal attack and defense strategies are and how much the attacker’s gains are limited by the optimal defense strategy when he employs the optimal attack strategy. In particular, we formulate a game theoretic problem as a two-player repeated game with perfect information, where the two players are the attacker and the defender. We believe that the repeated game is suitable for the analysis from the perspective of fully capturing the interactions between the attacker and the defender for a long period of time. In Table I, we summarize the notations that are frequently used in our long-term game theoretic analysis.

Suppose that the attacker deploys \( r \) replicas of a compromised node \( u \). Let \( N_c \) denote the number of compromised nodes in the network. Let \( \eta \) denote the total number of compromised nodes that can be received by \( u \)’s replicas. Let \( \psi_i \) denote the attacker’s replica placement strategy in the \( i \)th time slot. Specifically, \( \psi_i \) indicates how the attacker configures the distances between each pair of replicas during the \( i \)th time slot. We define \( \psi \) as attacker’s long-term replica placement strategy consisting of \( \psi_1, \psi_2, \ldots, \psi_i, \ldots \). Let \( f(\psi_i) \) denote the
fraction of \( \eta \) such that \( 0 < f(\psi_i) \leq 1 \) when \( \psi_i \) is used. Let 
\[ g(\psi_i) \] denote the ratio of the number of samples that exceed \( V_{\text{max}} \) to the total number of samples in the SPRT such that \( 0 < g(\psi_i) < 1 \) when \( \psi_i \) is used. Recall that \( p \) is the claim forwarding probability. Let \( h(\psi_i) \) denote the number of claim requests that are ignored or ones to which replicas respond with illegitimate claims as to not be detected when \( \psi_i \) is used. Then \( h(\psi_i) \) is given by

\[
h(\psi_i) = \max(g(\psi_i) f(\psi_i)(p\eta - 1) - \lceil \tau_i(f(\psi_i)(p\eta - 1)) \rceil + 1, 0)
\]

We configure the quarantine threshold \( \rho \) as a positive integer such that \( 1 \leq \rho \leq \rho_{\text{max}} = \lceil \max h(\psi_i) \rceil - 1 \). The rationale behind \( \rho_{\text{max}} \) is to quarantine replicas whose placement achieves the maximum value of \( h(\psi_i) \). Let \( I(h(\psi_i)) \) be a function of \( h(\psi_i) \). It is 0.0 if \( u \)'s replicas are under quarantine during the \( i \)th time slot; otherwise, it is 1.0. The quarantine period \( q \) is the number of quarantine time slots. If \( h(\psi_i) > \rho \), \( u \)'s replicas are quarantined from the \((i + 1)\)th to the \((i + q)\)th time slots and thus \( I(h(\psi_j)) = 0 \) for \( i + 1 \leq j \leq i + q \). For each time slot in which \( h(\psi_j) \) is more than \( \rho \), the quarantine period is incremented by one time slot. The pseudo-code for computing \( I(h(\psi_i)) \) is described as Function 1.

**Function 1:** Computation of \( I(h(\psi_i)) \)

- **INITIALIZE:** \( I(h(\psi_1)) = 1 \)
- **VARIABLES:** \( k, l \)
- **INPUT:** \( h(\psi_i), i \geq 1 \)
- **OUTPUT:** 0 or 1

if \( h(\psi_i) \leq \rho \) then
  if \( I(h(\psi_i)) \neq 0 \) then
    \( I(h(\psi_{i+1})) = 1 \)
  else
    if \( I(h(\psi_i)) \neq 0 \) then
      for \( k = 1; k \leq q; k++ \)
      \( I(h(\psi_{i+k})) = 0 \)
      \( l = i + q + 1 \)
    else
      \( I(h(\psi_i)) = 0 \)
      \( l = i + 1 \)
  end if
else
end if

We model the attacker’s payoff for a single time slot as the total number of nodes affected by the \( r \) replicas of \( u \) and the \( N_c \) compromised nodes. Specifically, we regard a node \( v \) as being affected by a replica of \( u \) if it sends a claim request to the replica and receives a valid claim, because \( v \) accepts the replica as its neighbor after receiving a valid and validating the claim. In the worst case, in which \( u \)'s replicas receive one claim request per neighbor, the number of nodes that are affected by \( u \)'s replicas is equivalent to the number of claim requests to which \( u \)'s replicas respond with legitimate claims as long as those replicas are neither detected nor quarantined. Since the attacker will get zero gain for the entire time period if the replicas are detected, he needs to ignore claim requests that contribute to detection or respond with illegitimate claims to those requests while losing the payoff corresponding to the number of ignored or illegitimately responded claim requests. However, if the number of ignored or illegitimately responded claim requests is more than quarantine threshold \( \rho \) during a time slot, the attacker will gain nothing during the quarantined time slots. Thus, he needs to limit the number of ignored or illegitimately responded claim requests in order to reduce his loss incurred by the quarantine defense strategy. By considering these factors, the number of nodes affected by \( u \)'s replicas during the \( i \)th time slot is represented as \( (f(\psi_i) - h(\psi_i)) I(h(\psi_i)) \) in the worst case. This expression indicates that the attacker’s gain is the number of claim requests to which \( u \)'s replicas respond with legitimate claims out of the received claim requests during the \( i \)th time slot as long as the number of ignored or illegitimately responded claim requests is at most \( \rho \) in the worst case. Also, we regard a node \( v \) as being affected by \( \rho \) compromised nodes if it is quarantined due to false DCN messages sent by \( \rho \) compromised nodes. By exploiting the property that quarantined nodes are put under quarantine for at least \( q \) time slots, the attacker can make at most \( q \) benign nodes be quarantined per time slot by using \( \rho \) compromised nodes. Accordingly, the total number of affected nodes by \( N_c \) compromised nodes during the \( i \)th time slot is at most \( q N_c / \rho \).

To model the attacker’s payoff for long period of time, we use the limit-of-means payoff as in [14]. This model is useful in the sense that attacker’s long term payoff is expressed in terms of the expected payoff per time slot and thus it can converge to a certain value when the number of time slots goes to infinity. We denote the attacker’s long term payoff by \( U(\rho, \psi) \), defined as:

\[
U(\rho, \psi) = \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} (f(\psi_i) - h(\psi_i)) I(h(\psi_i)) + \frac{q N_c}{\rho}
\]

where \( M \) is the number of time slots.

To explore the interactions between the attacker and the defender for long period of time, we formulate a minimax repeated game with \( U(\rho, \psi) \) as follows:

\[
\min_{\rho} \max_{\psi} U(\rho, \psi)
\]

In this game, the strategies of the attacker and the defender are \( \psi \) and \( \rho \), respectively. The attacker’s goal is to maximize his long term payoff \( U(\rho, \psi) \) by controlling the replica placement strategy \( \psi \). On the other hand, the defender’s goal is to minimize the maximum value of \( U(\rho, \psi) \) by controlling the quarantine threshold \( \rho \). The optimal strategies for the attacker and the defender are the ones that lead most closely to their respective goals. We now solve the above minimax optimization problem to find the optimal strategies for the attacker and the defender.

Let \( \psi^* \) and \( \rho_{\text{max}} \) be the attacker’s replica placement strategies, such that

\[
\arg \max_{\psi} [f(\psi_i) - h(\psi_i)] = \begin{cases} \psi^* & \text{if } 0 \leq h(\psi_i) \leq \rho \\ \psi_{\text{max}} & \text{if } h(\psi_i) > \rho \end{cases}
\]

Let us define \( w(\psi^*) \) such that \( h(\psi^*) = w(\psi^*) \rho \) and \( 0 \leq
$w(\psi^*) \leq 1$. We derive the optimal replica placement strategy $\psi^*$ in the following lemma.

Lemma 3.1: The argument of the maximum of $U(\rho, \psi)$ as a function of $\psi$ is $\psi^*$ under the condition that quarantine period $q \geq \frac{f(\psi^*)}{\eta - \hat{h}(\psi^*)} - 1$.

Proof: Let us denote $A(\psi_i) = f(\psi_i)\eta - h(\psi_i)$. We consider two cases, depending on $\rho$. In the case that $0 \leq h(\psi_i) \leq \rho$, $\sum_{i=1}^{M} A(\psi_i) \leq \sum_{i=1}^{M} A(\psi^*)$ holds. In the case that $h(\psi_i) > \rho$, $\sum_{i=1}^{M} A(\psi_i) \leq \sum_{i=1}^{M} A(\psi^*)$ holds. Recall that the quarantine period is $q$ time slots. If replica placement strategy $\psi^{max}$ is used in a time slot, replicas are under quarantine for the next $q$ time slots. Hence, the use of $\psi^{max}$ effectively results in the consumption of $q + 1$ time slots. Let us denote by $\varphi M$ the number of times that $\psi^{max}$ is used in $M$ time slots. By combining these two cases, we derive the following inequality on $\sum_{i=1}^{M} A(\psi_i)$ with $\varphi$ as follows:

$$\sum_{i=1}^{M} A(\psi_i)I(h(\psi_i)) \leq \varphi M A(\psi^{max}) + (M - \varphi(q + 1)M)A(\psi^*)$$

Therefore, the following inequality on $U(\rho, \psi)$ also holds:

$$U(\rho, \psi) \leq \varphi(A(\psi^{max}) - A(\psi^*)(1 + q)) + A(\psi^*) + \frac{qN_\rho}{\rho} \tag{9}$$

If $A(\psi^{max}) < A(\psi^*)(1 + q)$, the right side of Inequality 9 is a strictly decreasing function of $\varphi$ because its derivative with respect to $\varphi$ is less than zero and reaches its maximum value when $\varphi = 0$. Accordingly, $U(\rho, \psi) \leq A(\psi^*)(1 + q) + \frac{qN_\rho}{\rho}$ holds. If $A(\psi^{max}) = A(\psi^*)(1 + q)$, then $U(\rho, \psi) \leq A(\psi^*) + \frac{qN_\rho}{\rho}$ also holds. Hence, if $q \geq \frac{1}{\min(m, \rho^{max})} - 1 = \frac{1}{(\frac{\eta}{\rho} - \frac{\hat{h}(\psi^*)}{\rho}) - 1}$, then $U(\rho, \psi)$ reaches its maximum value when $\psi_i = \psi^*$ for all $i \geq 1$.

By Lemma 3.1, $\min(m, \rho^{max}), U(\rho, \psi)$ is equivalent to $\min_{\rho} U(\rho, \psi^*)$. Let us denote $V(\psi^*)$ such that $V(\psi^*) = \frac{qN_\rho}{\rho}$. We also denote $\nu_0, \nu_1, \nu_2$ such that $\nu_0 = \ln \frac{3\lambda}{\lambda_0} - \frac{\nu_1}{\lambda_0}$, $\nu_1 = \frac{1}{\rho'} - \frac{\alpha}{\rho}$, and $\nu_2 = \frac{1}{1 - \nu_1}$. Then $U(\rho, \psi^*)$ is given by

$$\rho^* = \begin{cases} 
1 & \text{if } \frac{qN_\rho}{\rho} \leq w(\psi^*) \\ \min(m, \rho^{max}) & \text{if } \frac{1}{(\frac{\eta}{\rho} - \frac{\hat{h}(\psi^*)}{\rho}) - 1} \leq w(\psi^*) \\ \rho^{max} & \text{if } w(\psi^*) = 0 
\end{cases}$$

for $m \geq 2$.

Proof: In the case that $w(\psi^*) = 0$, $U(\rho, \psi^*) = \frac{qN_\rho}{\rho} + f(\psi^*)\eta$ holds. Since $\frac{dU(\rho, \psi^*)}{dp} < 0$ on the interval $[1, \rho^{max}]$, $U(\rho, \psi^*)$ is decreasing function of $\rho$ on the interval $[1, \rho^{max}]$ and reaches its minimum value at $\rho = \rho^{max}$ on the interval $[1, \rho^{max}]$. Hence, $\rho^{*} = \rho^{max}$.

In the case that $0 < w(\psi^*) \leq 1$, $U(\rho, \psi^*) = \frac{qN_\rho}{\rho} + f(\psi^*)\eta - w(\psi^*)\rho$ holds. By using the property that

$$\tau_1(f(\psi_i)(p\eta - 1)) = \tau_1(f(\psi_i)(p\eta - 1)) + z(\psi_i)$$

such that $0 \leq z(\psi_i) < 1$ and the Equation 6, we can express $f(\psi^*)$ as follows:

$$f(\psi^*) = \frac{w(\psi^*)\rho + \frac{qN_\rho}{\rho} + z(\psi^*) - 1}{(g(\psi^*) - \frac{1}{\rho} - 1)} \tag{10}$$

By plugging $f(\psi^*)$ into $U(\rho, \psi^*)$, we have

$$U(\rho, \psi^*) = w(\psi^*)(V(\psi^*) - 1) + \frac{qN_\rho}{\rho} + \frac{\rho}{\rho^2} + z(\psi^*) - 1$$

Since the number of claim requests received by replicas is always larger than the number of claim requests to which they respond with legitimate claims as not to be detected when $\psi^*$ is used, $f(\psi^*)\eta > h(\psi^*)$ holds. Accordingly, $V(\psi^*) > 0$ also holds under the condition that $\frac{qN_\rho}{\rho} > 1$. If $V(\psi^*) = \frac{1}{(\frac{\eta}{\rho} - \frac{\hat{h}(\psi^*)}{\rho}) - 1}$, then $U(\rho, \psi^*) > 0$ holds because $0 < \frac{g(\psi^*)}{\rho^2} < 1$ and $0 < p < 1$. With the property $V(\psi^*) > 1$ under the condition that $\frac{qN_\rho}{\rho} > 1$, we consider three sub-cases as follows:

Sub-case 1: If $\frac{qN_\rho}{\rho} \leq w(\psi^*) \leq 1$, $\frac{dU(\rho, \psi^*)}{dp} = \frac{v(\psi^*)}{\rho} - \frac{qN_\rho}{\rho} > 0$ holds on the interval $[1, \rho^{max}]$.

Sub-case 2: If $\frac{qN_\rho}{\rho} > w(\psi^*) < \frac{qN_\rho}{\rho}$, then $U(\rho, \psi^*) \geq \frac{qN_\rho}{\rho} + \frac{\rho + z(\psi^*) - 1}{V(\psi^*)}$ holds. Since $\frac{dU(\rho, \psi^*)}{dp} = \frac{1}{\rho^2} > 0$ on the interval $[1, \rho^{max}]$ and $\min U(1, \psi^*) = \min U(2, \psi^*)$, $U(\rho, \psi^*)$ reaches its minimum value at $\rho = 1$. Hence, $\rho^* = 1$.

Sub-case 3: If $\frac{qN_\rho}{\rho} > \frac{1}{(\frac{\eta}{\rho} - \frac{\hat{h}(\psi^*)}{\rho}) - 1} \geq w(\psi^*) < \frac{qN_\rho}{\rho}$, then $U(\rho, \psi^*) \geq \frac{qN_\rho}{\rho} + \frac{qN_\rho}{\rho + \frac{1}{\rho - 1}}$ holds. Since $\frac{dU(\rho, \psi^*)}{dp} = \frac{dU(\rho, \psi^*)}{d\rho} > 0$ on the interval $[m + 1, \rho^{max}]$ and $\min U(m + 1, \psi^*) = \min U(m, \psi^*)$, $U(\rho, \psi^*)$ reaches its minimum value at $\rho = m$. Since $\rho$ should be configured to be at most $\rho^{max}$, $\rho^{*} = \min(m, \rho^{max})$.

Theorem 3.1: If we set the quarantine period $q$ such that $q \geq \frac{f(\psi^*)\eta - h(\psi^*)}{\rho^2} - 1$, the strategies $\psi^*$ and $\rho^*$ are optimal for the attacker and defender, respectively.

Proof: By Lemma 3.1, the arguments of the minima of $U(\rho, \psi)$ are $\psi^*$ and $\rho^*$ under the condition that $q \geq \frac{f(\psi^*)\eta - h(\psi^*)}{\rho^2} - 1$. Therefore, the optimal strategies of the attacker and defender are $\psi^*$ and $\rho^*$ if $q$ is at least $\frac{f(\psi^*)\eta - h(\psi^*)}{\rho^2} - 1$.

Now we examine the characteristics of the functions $f(\psi_i), g(\psi_i)$, and $h(\psi_i)$ in terms of $\psi_i$. Assume that the average distance between a pair of replicas in $\psi_i^a$ is less than 1.0 in $\psi^t$. As the distance between a pair of replicas decreases, the overlapped neighborhood areas between a pair of replicas increases; accordingly the total number of affected nodes decreases. Hence, $f(\psi_i^a) < f(\psi_i^t)$. Moreover, the decrease of the distance between a pair of replicas leads to a reduction of the measured speed of replicas, and thus the likelihood that a sample exceeds $V^{\max}$ in the SPRT falls off. Hence, both $g(\psi_i^a)$ and $h(\psi_i^a)$ are less than $g(\psi_i^t)$ and $h(\psi_i^t)$.

Next we investigate the limitation on the attacker’s gain when the defender and attacker adhere to their optimal strate-
the compromised nodes only affect $qN$ number of claims. When $0 < w(\psi^*) < 1$, $\rho^*$ is set to a value between 1.0 and $\rho^{\text{max}}$. As $w(\psi^*)$ goes to zero and one, $\rho^*$ goes to $\rho^{\text{max}}$ and 1.0 and accordingly $h(\psi^*) = w(\psi^*)\rho^*$ goes to zero and one, respectively. Therefore, $h(\psi^*)$ takes on larger values as $w(\psi^*)$ is further away from zero and one. Under this intuition, we denote the maximum values of $w(\psi^*)$ and $\rho^*$ by $\phi_1$ and $\phi_2\rho^{\text{max}}$ such that $0 < \phi_1, \phi_2 < 1$, respectively. Hence, the maximum value of $h(\psi^*)$ is denoted by $\phi_1\phi_2\rho^{\text{max}}$. Since the value of $h(\psi^*)$ is limited by $\phi_1\phi_2\rho^{\text{max}}$, $f(\psi^*)\eta - h(\psi^*)$ is also limited by replica placement strategy $\psi^*$ such that $h(\psi^*) = \phi_1\phi_2\rho^{\text{max}}$. We therefore have only $\sum_{n=1}^{\text{max}} \frac{\phi_1\phi_2\rho^{\text{max}}}{\phi_1\phi_2\rho^{\text{max}}} = N_c$ compromised nodes.

In all three cases, we see that attacker has limited benefit from employing replicas and compromised nodes when defender and attacker follow their optimal strategies.

### C. Performance Analysis

We now analyze the performance of our scheme in terms of communication, computation, and storage overheads.

1) Communication Overhead: We first describe how many observations on an average are required for the base station to make a decision as to whether a node has been replicated or not. Then we will present the communication overhead of our scheme.

Let $n$ denote the number of samples to terminate the SPRT. Since $n$ varies with the types of samples, it is treated as a random variable with expected value $E[n]$. According to [15], $E[n]$ is obtained as follows:

$$E[n] = \frac{E[L_0]}{E[L_0] P_{R(S_i/H_0)} - E[L_1]/P_{R(S_i/H_1)}}$$

(11)

From this equation, we compute the expected numbers of $n$ conditioned on the hypotheses $H_0$ and $H_1$ as follows:

$$E[n|H_0] = \frac{(1 - \alpha') \ln \frac{\beta'}{\beta'} \ln \frac{1 - \alpha'}{\alpha'} + \alpha' \ln \frac{1 - \beta'}{1 - \alpha'}}{\lambda_0 \ln \frac{1}{\lambda_0} + (1 - \lambda_0) \ln \frac{1 - \beta'}{1 - \alpha'}}$$

$$E[n|H_1] = \frac{\beta' \ln \frac{\beta'}{\beta'} + (1 - \beta') \ln \frac{1 - \beta'}{1 - \alpha'}}{\lambda_1 \ln \frac{1}{\lambda_1} + (1 - \lambda_1) \ln \frac{1 - \beta'}{1 - \alpha'}}$$

(12)

We study how $E[n|H_0]$ and $E[n|H_1]$ are affected by the values of $\lambda_0$ and $\lambda_1$. As shown in Figures 2 and 3, $E[n|H_0]$ and $E[n|H_1]$ tend to increase in proportion to $\lambda_0$ when $\lambda_1$ is fixed to 0.7 and 0.9, respectively. This implies that a small value of $\lambda_0$ contributes to detecting replicas with a small number of claims. When $\lambda_0$ is fixed, $E[n|H_0]$ and $E[n|H_1]$ for the case of $\lambda_1 = 0.7$ are larger than the corresponding values for the case of $\lambda_1 = 0.9$. This means that a larger value of $\lambda_1$ reduces the number of claims required for benign node decision and replica detection.

Now let us compute the communication overhead of our scheme. We define the communication overhead as the average number of claims that are sent or forwarded by nodes in the network. Each time a mobile node $u$ receives $b$ claim requests on an average at a location, it sends an average of $b \times \rho$ claims to the base station, where $p$ is the probability that the claim is forwarded to the base station. Let us consider the worst case scenario in which every mobile node receives $b$ claim requests at a location and sends $b \times p$ claims to the base station at the same time. Since the average hop distance between two randomly chosen nodes is given by $O(\sqrt{N})$ [11] where $N$ is the total number of sensor nodes, the communication overhead in the worst case will be $O(b \times \rho \times N \times \sqrt{N})$. Each node’s $b$ claim requests contain the same location information $L$. Indeed, $O(1)$ claim per location $L$ is enough for the base station to perform the replica detection. In this sense, $b \times p$ can be reduced to $O(1)$ by setting $p$ to $k\frac{1}{b} = O(1)$, for some constant $k$. In this configuration, the probability that a node’s claim is forwarded by at least one neighbor to the base station is computed as $1 - (1 - p)^k$. For example, this probability becomes 0.961 when $b = 20$, $k = 3$, and $p = k\frac{1}{b} = 0.333$, ensuring that the base station receives a node’s claim with high probability. Thus, the communication overhead in the worst case can be rewritten as $O(N\sqrt{N})$.

2) Computation and Storage Overhead: We define computation and claim storage overhead as the average number of public key signing and verification operations per node and the average number of claims that needs to be stored by a node, respectively.

Each time a mobile node receives $b$ claim requests on an average at a location, it needs to perform $b$ signature generation operations. Similarly, each time a mobile node sends $b$ claim requests on an average at a location, it needs to verify up to $b$ signatures. In the worst case, every mobile node sends $b \times p$ claims to the base station at the same time and the base station thus needs to verify up to $b \times \rho \times N$ signatures. If $p$ is set to $k\frac{1}{b}$, the base station will verify up to $k \times N$ signatures on average in the worst case.

The base station stores location claims in order to perform the SPRT, whereas the sensor nodes do not need to keep its own or other nodes’ claims. Thus, we only need to compute the number of claims that are stored by the base station. In the SPRT, a sample $o_i$ is obtained from two consecutive location claims of node $u_i$, namely $C_{u_i}^{i-1}$ and $C_{u_i}^i$. Once a sample $o_i$ is obtained, the previous location claim $C_{u_i}^{i-1}$ is discarded and current location claim $C_{u_i}^i$ is maintained by the base station. This process is repeated until the SPRT is terminated. Hence, the base station needs to store only one claim per node, so at most $N$ claims are required to be stored in the base station.

### IV. Simulation Study

In this section, we will first describe the simulation environment used to evaluate our scheme and then present our experimental results.
A. Simulation Environment

We simulated the proposed mobile replica detection scheme in a mobile sensor network with the help of the ns-2 network simulator. In our simulation, 500 mobile sensor nodes are placed within a square area of 500 m × 500 m.

We use the Random Waypoint Mobility (RWM) model to determine mobile sensor node movement patterns. In particular, to accurately evaluate the performance of the scheme, we use the RWM model with the steady-state distribution provided by the Random Trip Mobility (RTM) model [1]. In the RWM model, each node moves to a randomly chosen location with a randomly selected speed between a predefined minimum and maximum speed. After reaching that location, it stays there for a predefined pause time. After the pause time, it then randomly chooses and moves to another location. This random movement process is repeated throughout the simulation period. We use code from [10] to generate RWM-based movements model with a steady-state distribution.

All simulations were performed for 1000 simulation seconds. We fixed a pause time of 20 simulation seconds and a minimum moving speed of 1.0 m/s of each node. Each node uses IEEE 802.11 as the medium access control protocol in which the transmission range is 50 m. We set both the user-configured false positive threshold \( \alpha' \) and the false negative threshold \( \beta' \) to 0.01.

To emulate the speed errors caused by the inaccuracy of time synchronization and localization protocols, we modify the measured speeds with maximum speed error rate \( \gamma \). Specifically, we take speed \( s \) measured using perfect time synchronization and localization protocols and generate speed \( s' \) selected uniformly at random from the range \([s-s\gamma, s+s\gamma]\). We evaluated the scheme with \( \gamma \) values of 0.01, 0.1, and 0.2. We set \( \lambda_0 \) and \( \lambda_1 \) in accordance with \( \gamma \) and \( V_{\text{max}} \) as shown in Table II, where \( L, M, H, \) and \( V \) indicate low

<table>
<thead>
<tr>
<th>( \gamma )</th>
<th>0.01</th>
<th>0.1, 0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility Rate</td>
<td>( \lambda_0 )</td>
<td>( \lambda_1 )</td>
</tr>
<tr>
<td>( L )</td>
<td>0.1</td>
<td>0.95</td>
</tr>
<tr>
<td>( M )</td>
<td>0.05</td>
<td>0.9</td>
</tr>
<tr>
<td>( H )</td>
<td>0.01</td>
<td>0.8</td>
</tr>
<tr>
<td>( V )</td>
<td>0.2</td>
<td>0.85</td>
</tr>
<tr>
<td>( H )</td>
<td>0.15</td>
<td>0.8</td>
</tr>
</tbody>
</table>

\( V_{\text{max}} = 10 \) m/s, moderate \( V_{\text{max}} = 20, 40 \) m/s, high \( V_{\text{max}} = 60 \) m/s, and very high \( V_{\text{max}} = 80, 100 \) m/s mobility rates, respectively. When we consider robotic vehicular platforms, low (no more than 36 km/hour) and moderate (no more than 72 or 144 km/hour) mobility rates may be suitable. High (no more than 216 km/hour) and very high (no more than 288 or 360 km/hour) mobility rates may be suitable for modeling autonomous aircraft.

The rationale behind the general configurations of \( \lambda_0 \) and \( \lambda_1 \), which are used in the SPRT, is discussed in Section III-C. As shown in Table II, these two parameters are modified in inverse proportion to changes in the mobility rate. The main reason for these configurations is because both mobility and speed error contribute to reduce the chance that a mobile node’s speed exceeds \( V_{\text{max}} \).

In our simulation, we consider two cases: mobileReplica and staticReplica. In the mobileReplica case, we use one benign node and one compromised node along with its replica as claim generators. Furthermore, these three nodes’ initial placements are randomly chosen and their movements are randomly determined by the RWM model with a steady-state distribution. In the staticReplica case, we use one compromised node along with its replica as claim generators. These two nodes do not move, as we fix their locations to the initial placements. By studying the staticReplica case, we can investigate how the distance between the compromised node and its replica affects the replica detection capability. The staticReplica case represents a strategic attacker and effectively the worst case for detection. The attacker keeps his nodes close together and immobile to lower the chance of detection. As analyzed in Section III-B, this also limits the attackers’s effectiveness. In all scenarios, we assume that all claims that have been forwarded to the base station reach it without any loss. We repeated each simulation scenario 1000 times in such a way that the mobile nodes are initially placed in a different random location each time.
Fig. 5. Probability distribution of the number of claims when $\gamma = 0.1$.

Fig. 6. Average number of claims vs. D when $V_{max} = 10$ m/s.

Fig. 7. Average number of claims vs. D when $V_{max} = 20$ m/s.

Fig. 8. Average number of claims vs. D when $V_{max} = 40$ m/s.

Fig. 9. Average number of claims vs. D when $V_{max} = 60$ m/s.

Fig. 10. Average number of claims vs. D when $V_{max} = 80$ m/s.

Fig. 11. Average number of claims vs. D when $V_{max} = 100$ m/s.

Fig. 12. Prob. distribution of the number of claims ($V_{max}=10$ m/s).

Fig. 13. Prob. distribution of the number of claims ($V_{max}=100$ m/s).

B. Simulation Results

We use the following metrics to evaluate the performance of our scheme:

- **Number of Claims** is the number of claims required for the base station to decide whether a node has been replicated or not.
- **False Positive** is the error probability that a benign node is misidentified as a replica node.
- **False Negative** is the error probability that a replica node is misidentified as a benign node.

For each execution, we obtain each metric as the average of the results of the SPRTs that are repeated. Note that the SPRT will be terminated if it decides that the claim generator has been replicated. The average of the results of 1000 executions is presented here.

In the experiments, the average number of claim requests, $b$, was measured between 17 and 20, depending on $V_{max}$. We associate the configuration of claim forwarding probability $p$ with $b$. Specifically, $p$ is configured to 0.05 in order to set $b \times p$ to one when $b$ is assumed to be 20 at all $V_{max}$. The rationale behind this configuration is to make sure that one claim per location is forwarded to the base station on average.

1) Mobile Replica Results: In the mobileReplica case, we investigate the false positive and false negative rates and the number of claims while increasing $V_{max}$ in the range from 10 m/s to 100 m/s. The results for the mobileReplica case are summarized as follows.

First, both false positives and false negatives were below 0.013 at all speed error rates and mobility rates. Specifically, the lowest and highest false positives were measured as 0.0, 0.007 when $\gamma = 0.01, 0.1$ and 0.003, 0.013 when $\gamma = 0.2$, respectively. We also observed that there were zero false negatives when $\gamma = 0.01$, while the lowest and the highest false negative rates were 0.0, 0.006 when $\gamma = 0.1, 0.2$, respectively. Thus, the replica was detected with at least probability of 0.994 and the benign node was misidentified as a replica with at most probability of 0.013 at all speed error rates and mobility rates.

Second, the results of the average number of claims are shown in Figure 4. We present the results for two cases. One is that the claim generator is a benign node and the SPRT decides that this node is benign. We denote this case by trueNegative
in Figure 4. The other case is that the claim generators consist of a compromised node and its replica, and the SPRT decides that these nodes are a compromised node and its replica. We denote this case by truePositive in Figure 4.

In the trueNegative case, the average number of claims reaches a maximum of 5.03 when $V_{max} = 80$ m/s and $\gamma = 0.2$. In the truePositive case, the average number of claims reaches a maximum of 9.089 when $V_{max} = 10$ m/s and $\gamma = 0.2$. Thus, the base station reaches correct decisions with a few claims in both cases. Moreover, we see that the average at $\gamma = 0.1, 0.2$ is higher than that at $\gamma = 0.01$ in both cases. This indicates that a substantial increase in the speed error rate leads to a rise in the average number of claims.

We also observe that the average number of claims tends to slightly increase and decrease as mobility rate rises in the case of trueNegative and truePositive, respectively. We infer from this observation that a rise in mobility increases the chance that the speed of a benign node is erroneously measured to be over $V_{max}$, thus delaying the test from moving toward $H_0$. On the other hand, a rise in mobility leads to a reduction in the chance that the replicated node generates claims containing the same location but different time, and thus expedites moving the test toward $H_1$.

Finally, Figure 5 shows the probability distribution of the number of claims in the case of truePositive when $\gamma = 0.1$. For this distribution, we examine two scenarios: low mobility ($V_{max} = 10$ m/s) and very high mobility ($V_{max} = 100$ m/s). A total of 76.3% and 73.65% of the cases fall in the range from four to nine claims in the case of low and very high mobility rates, respectively. This implies that, in most cases, the number of claims is less than or close to the average and thus the SPRT detects replicas with at most nine claims in most cases.

2) Static Replica Results: In the staticReplica case, we investigate the same metrics as in mobileReplica case. We fix the positions of the compromised node $u$ and its replica $u'$, but vary their initial positions such that the distance $D$ between them varies in the range from $\frac{D_{max}}{2}$ to $2D_{max}$, where $V_{max} = D_{max}/s$ and the range of $V_{max}$ is from $10$ m/s to $100$ m/s. We do this to determine the ability of our scheme to detect replica nodes that are relatively close together, which is important when detection is based on speed.

First, false negative rates were measured as 0.008 for $D = 20$ m, $V_{max} = 40$ m/s, and $\lambda = 0.1, 0.2$. They were 0.0 for all other speed error rates, mobility rates, and $D$. Accordingly, $u$ and $u'$ were detected with at least 0.992 probability in all cases. Every mobile node checks whether to send claim requests to $u$ or $u'$ every 0.5 seconds. Thus, a mobile node’s claim request time period is at least 0.5 seconds. Under this claim request time period, we infer from the high detection rate that the inter-arrival time between the claim requests to $u$ and $u'$ is highly likely to be less than $\frac{D}{V_{max}}$. Subsequently, this implies that attacker needs to configure the distance $D$ in less than $\frac{D_{max}}{2}$ under the above claim request time period configuration to have a reasonable chance that $u$ and $u'$ are not detected.

Second, Figures 6, 7, 8, 9, 10, and 11 show the average number of claims when $V_{max} = 10, 20, 40, 60, 80, 100$ m/s, respectively. In the truePositive case, the average number of claims is below 10.5 at all mobility rates $V_{max}$ and distances $D$. Hence, the base station detects $u$ and $u'$ with a reasonable number of claims. In terms of the effect of $\gamma$ on the average number of claims, we see that the substantial increase of $\gamma$ from 0.01 to 0.1, 0.2 contributes to a rise in the average number of claims. However, the number of claims remains small. In terms of the affect of $D$ on the average number of claims, we observe that the rise of $D$ from $\frac{D_{max}}{2}$ to $D_{max}$, $2D_{max}$ results in an increase in the average number of claims to detection. We infer from this observation that larger values of $D$ defer the occurrence that the inter-arrival times between the claim requests to $u$ and $u'$ exceed $\frac{D}{V_{max}}$, leading to delay in moving the test toward $H_1$.

Finally, Figures 12 and 13 show the probability distribution of the number of claims in truePositive cases when $\gamma = 0.1$ and $V_{max}$=10 m/s, $\gamma = 0.1$ and $V_{max}$=100 m/s, respectively. For each case, we examine two scenarios: short distance $D = \frac{D_{max}}{2}$ and long distance $D = 2D_{max}$. In case of $V_{max}$=10 m/s, the average number of claims ($\mu'$) and standard deviation ($\sigma'$) in the short and long distance scenarios are 7.207 and 1.792, 8.425 and 3.762, respectively. In case of $V_{max}$=100 m/s, $\mu'$ and $\sigma'$ in the short and long distance scenarios are 7.112 and 3.854, 9.905 and 6.899, respectively. The fraction of the number of claims up to $\mu' + \sigma'$ is at least 86% and 82% in the case of $V_{max}$=10 m/s and $V_{max}$=100 m/s, respectively. This means that in most cases, the number of claims does not exceed $\mu' + \sigma'$.

In short and long distance scenarios, a total of 91.8% and 80.6% of the case of $V_{max}$=10 m/s fall in the range from five to nine claims, respectively. In addition, a total of 80.1% and 61.1% of the cases of $V_{max}$=100 m/s fall in the range from four to nine claims in short and long distance scenarios, respectively. Thus, detection occurs quickly with few claims in most cases in all the scenarios we examined.

V. RELATED WORK

The first work on detecting replica node attacks is due to Parno et al. [11], who proposed randomized and line-selected multicast schemes to detect replicas in static wireless sensor networks. In those two schemes, nodes report location claims that identify their positions and attempt to detect conflicting reports that signal one node in multiple locations. Conti et al. [3] proposed a scheme to enhance the line-selected multicast scheme of [11] in terms of replica detection probability, as well as storage and computation overheads by using trusted random values. Ho et al. [6] proposed several schemes for distributed detection of replica nodes that take advantage of group deployment knowledge to reduce the communication, computation, and storage overheads required for replica detection and improve on the replica detection capability of the line-selected scheme of [11]. Xing et al. [17] proposed a fingerprint-based replica node detection scheme. In this scheme, nodes report fingerprints, which identify a set of their neighbors, to the base station. The base station performs replica detection by using the property that fingerprints of replicas conflict each other.
However, none of these solutions is suitable for replica node detection in mobile sensor networks. If the schemes in [3], [6], [11] are used in mobile sensor networks, sensor nodes' location claims will be continuously changed in accordance with their movements, and thus location claims from the same benign node will always conflict each other. Similarly, if the scheme in [17] is used in mobile sensor networks, mobility will continuously make nodes have different fingerprints, and thus fingerprints of the same benign node will conflict each other.

Recently, Yu et al. [18] proposed schemes to detect node replica attacks in mobile sensor networks. The key idea of [18] is to detect mobile replicas by leveraging the intuition that the number of mobile nodes encountered by mobile replicas in a time interval is more than the number encountered by a benign mobile node. The worst-case communication and storage overheads in our scheme are computed as $O(N\sqrt{N})$ and $O(1)$, respectively; whereas these values are respectively $O(N^2)$ and $O(N)$ when the schemes in [18] are used. Therefore, our scheme works with less overhead than those in [18]. The main strength of [18] is that it detects mobile replicas in fully distributed manner, while our scheme relies on the base station for mobile replica detection. However, replicas can evade this detection technique by carefully controlling the number of encounters each replica has with other nodes. The attacker can selectively uses its encounters to maximize the effectiveness of the attacks it is trying to mount with the replica nodes. Since this puts a limitation on the attacker, it remains to be studied whether the detection scheme is enough to deter effective replica attacks.

VI. CONCLUSIONS

In this paper, we have proposed a replica detection scheme for mobile sensor networks based on the Sequential Probability Ratio Test (SPRT). We have analytically demonstrated the limitations of attacker strategies to evade our detection technique. In particular, we first showed the limitations of a group attack strategy in which the attacker controls the movements of a group of replicas. We presented quantitative analysis of the limit on the amount of time for which a group of replicas can avoid detection and quarantine. We also modeled the interaction between the detector and the adversary as a repeated game and found a Nash equilibrium. This Nash equilibrium shows that even the attacker’s optimal gains are still greatly limited by the combination of detection and quarantine. We performed simulations of the scheme under a random movement attack strategy in which the attacker lets replicas randomly move in the network and under a static placement attack strategy in which he keeps his replicas from moving to best evade detection. The results of these simulations show that our scheme quickly detects mobile replicas with a small number of location claims against either strategy.

VII. ACKNOWLEDGEMENTS

We sincerely thank the anonymous reviewers for insightful comments which helped us improve the quality of the manuscript significantly. Jun-Won Ho was supported by IT Scholarship Program under the supervision of IITA (Institute for Information Technology Advancement) and MIC (Ministry of Information and Communication) in Republic of Korea. This work is partially supported by NSF Grant IIS-0326505, CNS-0721951 and CNS-0916221 and AFOSR Grant A9550-08-1-0260. The work of S. K. Das is also supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES


Jun-Won Ho received his B.S. degree from the Department of Computer Science at Yonsei University, Seoul, South Korea, M.S. degree from the Department of Electrical and Computer Engineering at University of Wisconsin at Madison, Ph.D. degree from the Department of Computer Science at the
University of Texas at Arlington in May, 2010. His dissertation work focuses on the detection and prevention of wide-spread node compromise in wireless sensor networks. His current research interests include security in wireless sensor networks.

Matthew Wright is an assistant professor at the University of Texas at Arlington. He graduated with his Ph.D from the Department of Computer Science at the University of Massachusetts in May, 2005, where he earned his M.S. in 2002. His dissertation work addresses the robustness of anonymous communications. His other interests include intrusion detection, security and privacy in mobile and ubiquitous systems, and the application of incentives to security and privacy problems. Previously, he earned his B.S. degree in Computer Science at Harvey Mudd College. He is a recipient of the Outstanding Paper Award at the 2002 Symposium on Network and Distributed System Security.

Sajal K. Das is received the B.S. degree in computer science from Calcutta University, Kolkata, India, in 1983, the M.S. degree in computer science from the Indian Institute of Science, Bangalore, India, in 1984, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, in 1988. He is currently with the Department of Computer Science and Engineering, the University of Texas at Arlington, where he is a Distinguished Scholar Professor of Computer Science and Engineering and the Founding Director of the Center for Research in Wireless Mobility and Networking (CReWMaN). He is also currently a Program Director with the Division of Computer Networks and Systems, National Science Foundation (NSF). He is the author of more than 400 published papers and more than 35 invited book chapters. He is a coauthor of the books Smart Environments: Technology, Protocols, and Applications (Wiley, 2005) and Mobile Agents in Distributed Computing and Networking (Wiley, 2009). He is the Founding Editor-in-Chief of Elsevier's Pervasive and Mobile Computing journal. He is an Associate Editor of ACM/Springer Wireless Networks, Journal of Parallel and Distributed Computing, and Journal of Peer-to-Peer Networking. His current research interests include wireless and sensor networks, mobile and pervasive computing, smart environments, pervasive security, applied graph theory, and game theory. In addition to several best paper awards, he is a recipient of the IEEE Computer Society Technical Achievement Award (2009) and the IEEE Region 5 Outstanding Engineering Educator Award (2009). He is an Associate Editor of IEEE Transactions on Mobile Computing and IEEE Transactions on Parallel and Distributed Systems. He has served as the General and Technical Program Chair of numerous conferences.