Location Verification Based Defense Against Sybil Attack In Sensor Networks

Debapriyay Mukhopadhyay and Indranil Saha

Honeywell Technology Solutions Lab Pvt. Ltd.
151/1, Doraisanipalya, Bannerghatta Road,
Bangalore 560 076, India
{debapriyay.mukhopadhyay, indranil.saha}@honeywell.com

Abstract. Security is a major concern for a large fraction of sensor network applications. Douceur first introduced the notion of sybil attack [4], where a single entity(node) illegitimately presents multiple identities. As the nodes in sensor networks can be physically captured by an adversary, sybil attack can manifest in a severe form leading to the malfunction of basic operational protocols including routing, resource allocation and misbehavior detection. In this study, we propose a location verification based defense against sybil attack for sensor network where we assume that the network is consisted of static sensor nodes. We report quantitatively about the probability of not being able to detect sybil attack. This probability is indicative of the usefulness of our proposed protocol.

Keywords: Sybil attack, Security, Wireless Sensor Network, Triangulation, Location Verification.

1 Introduction

Sensor networks are now being widely deployed in planned or ad hoc basis to monitor and protect different targeted infrastructures including life-critical applications such as wildlife monitoring, military target tracking, home security monitoring and scientific exploration in hazardous environments. The criticality of a large subset of applications triggers the need for providing adequate security support for them. Unlike in general data networks, the nodes of sensor networks may be physically captured by an adversary and thus can induce different modes of harmful attacks in addition to active and passive eavesdropping. This typical feature also makes the design of cryptographic primitives for sensor networks extremely challenging.

Douceur first introduced the notion of sybil attack [4], where a single entity illegitimately presents multiple identities. Physically captured nodes claiming superfluous misbehaving identities could control a substantial fraction of the system leading to malfunction of basic operational protocols including routing, resource allocation and misbehavior detection. An excellent taxonomy of sybil attacks in sensor networks and their detrimental effects are presented by Newsome *et. al.* in [8], along with some defense mechanisms. In their work, they have provided a definition of simultaneous sybil attack where an attacker tries to have his sybil identities all participate in the network at once.

Sybil attack could be prevented if each honest identity possesses an unforgeable certificate issued by some trusted Certifying Authority(CA) and it is mandated to produce that certificate as a proof of authenticity before the identity takes part in any network activity. This condition implies that for inducing sybil attack the adversary has to necessarily forge valid certificates. But since sensor nodes are resource constrained devices, so computationally expensive public key cryptography based certification schemes are not suitable to be applicable in sensor networks.

It has also been mentioned in [8] that location verification could be a promising approach to defend sybil attack. In this work, we aim to provide such a solution for defending simultaneous sybil attack. Our solution does not require to verify the exact physical position of a node, rather it works out by securely verifying whether the physical position of the node is within a region. The region is defined in terms of a new functional for planar triangulation which we have come up with and we call it as *Inner Core* of a triangulation. A lot of theoretical and algorithmic questions come along with this new functional, but in this paper we just try to show how this functional could be of use in defending sybil attack rather than delving into solving those problems. Lastly, our solution is mainly targeted for those sensor network based applications where it is required to deploy sensor nodes in a planned manner.

We organize our paper as follows. Section 2 describes prior art and also briefly reviews the merits and demerits of each. In Section 3, we formally define the problem by clearly mentioning network assumptions and security goals. Section 4 defines the new functional Inner Core for planar triangulation. In Section 5, we describe our solution and also provide a formal analysis of its security. Lastly, Section 6 concludes this work along with future directions of research.

2 Related Work

Sybil attack was first introduced by Douceur in [4], wherein a direct validation method of a node's identity based on resource testing was proposed. The basic idea of the scheme is to estimate the resource (*e.g.,* computation, storage and communication) associated with each identity and thereby deciding whether each identity possesses a dedicated hardware piece. Scheme proposed in their work applies for general P2P networks and is not suitable to be applicable in sensor networks where an adversary may bring in very powerful devices (in terms of computation, storage and communication) to defeat the scheme. Karlof *et. al.* analyzed different attacks including sybil attack in [7] for wireless sensor network and described some countermeasures against them. In their approach, each node is provided with a unique symmetric key which it shares with a trusted base station. Two nodes then can verify each other's identity through establishment of a shared key, via the base station, using symmetric Needham-Schroeder-like [2] protocol. The solution thus relies on the existence of a trusted third party. Again, there are some attacks [1] against Needham-Schroeder-like protocol in which case the proposed solution fails.

Wang *et. al.* [5] introduced the concept of trust graph in mobile ad hoc network, which facilitates in establishing trust relationship between communicating nodes and considers the possibility of having heterogeneous certifying authorities (CAs). Assumption here is that if a certifying authority CA_1 trusts another certifying authority CA_2, then CA_1 also trusts identities certified by CA_2. It is interesting to note here that this assumption and mechanism can safeguard against sybil attack as long as none of the CAs is compromised. Their scheme also demands each node to have moderately high storage and computational capability and also charges high communication cost and thus remains unsuitable for sensor networks. Newsome *et. al.* in their work [8] established a taxonomy of different kinds of sybil attack and provided two methods based on radio resource testing and random key predistribution to verify whether a node's identity is a sybil identity. Though their random key pre-distribution based scheme is very promising, but still its not mature enough to conceive the notion of certificate in the symmetric key domain. The scheme also has limitation in a typical scenario where the nodes of the sensor network come from different vendors.

Kong *et. al.* proposed a public key cryptography based distributed threshold certification scheme [3] which establishes trust relationship between communicating nodes via unforgeable, renewable and globally verifiable certificates carried by each node in the network. Its not explicitly mentioned that their work can stand against sybil attack as otherwise attacker has to guess valid certificate of the claimed identity. But, their work is based on public key cryptography and targeted to meet the needs of wireless ad hoc networks. It does not readily suit well in resource constrained sensor network architecture. Zhang *et. al.* in [10] have proposed an identity certificate based scheme to defend against sybil attack in sensor networks. Their method associates each node's identity with an unique identity certificate, where Merkle hash tree has been used as the basic means of computing identity certificates. Main drawback of the scheme is that it is not scalable as it does not allow nodes to join the network on the fly because of huge computational overhead. Method also requires a large number of messages getting exchanged to build a trust between a pair of nodes and also it can not stand against sybil attack launched by colluding nodes.

3 Problem Definition

Network Assumptions: The network is consisted of a large number of sensor nodes and is deployed by a single authority. We are considering here those sensor network based applications where it is required to deploy sensor nodes in a planned manner. We also assume the presence of a powerful setup server which configures the sensor network and is aware of the locations of all the sensor nodes being deployed. Once deployed, each node is static and can be thought of as placed in a plane with each node having a distinct position where it is placed / deployed. We also allow new nodes to join the network on the fly. But since its a planned deployment, so we only allow joining of new nodes within the convex hull of initial set of deployed nodes. This is not a very unrealistic assumption since for a planned deployment its not difficult to know a priori the region in which

to deploy the sensor nodes and accordingly we can then deploy the initial set of sensor nodes. All the sensor nodes are capable of communicating using both radio frequency (RF) and sound channel and can only directly communicate with a limited number of neighboring nodes.

We also assume that both the transmission channels are perfectly secure, that is, any message being sent over any of these channels reaches the destination as unintercepted. All the sensor nodes trust the setup server and assume its behavior to be perfectly fare. Unlike in general data networks, nodes of a sensor network are susceptible to physical capture by an adversary and then can control them to attack the network. But the fraction of them will only be a small percentage of the overall network.

Security Goals: Our goal in this paper is to provide a mechanism to safeguard against simultaneous sybil attack where a malicious node simultaneously claims many identities of itself to defeat some of the well known protocols like data aggregation, routing, etc. An attacker who launches a simultaneous sybil attack will attempt to position the sybil identities in strategic locations of the network in order to defeat the above protocols. We thus choose to verify securely the position of any new node that pops up in the network, which in turn can act as a mechansim to defend against sybil attack.

4 Inner Core: A New Functional for Planar Triangulation

Definition 1 *Inner Core of a triangle (Figure 1) T with $v_i (i = 1, 2, 3)$ as vertices is defined as,*

$$IC(T) = \{\cap_{i=1}^{3} Disk(V_i, l_i)\} \cap T,$$

where l_i = min {Length of the sides of the triangle T incident on V_i}, and $Disk(V_i, l_i)$ is the circular region with V_i as it's center and l_i as its radius.

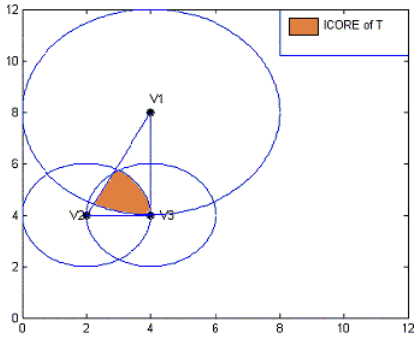

Fig. 1. Inner Core of Triangle T

Inner Core of triangulation Δ of a set $S \subset R^2$ of planar points is defined as the union of the inner cores of its constituent triangles, i.e.,

$$IC(\Delta) = \bigcup_{T \in \Delta} IC(T).$$

For a set S of planar points, the set $\mathcal{F}$ of all triangulations becomes exponential in size with the number of planar points. So, a natural and obvious question is to find out the triangulation for which Inner Core gets maximized, i.e., to find out $\Delta \in \mathcal{F}$ for which the area of $ConvH(S) - IC(\Delta)$ is minimized, where $ConvH(S)$ denotes the convex hull of the set of planar points S.

Delaunay triangulation D of a set of points $S \subset R^2$ forming a regular triangular lattice coincides with the lattice itself and hence $ConvH(S) - IC(D) = \phi$ and thus maximizes Inner Core. Getting started with this observation, we haven't been able to either prove or disprove that Inner Core achieves its maximum for Delaunay triangulation and hence we make the following conjecture.

Conjecture 1 *The functional $IC(\Delta)$ of a set of planar points S achieves its maximum if and only if Δ is the Delaunay triangulation of S.*

5 Protocol

In this section we describe the applicability of Inner Core in preventing sybil attack. We assume the presence of a software agent who is aware of the locations of all the deployed sensor nodes. Agent can get to know about the location of a sensor node either by consulting with the setup server or when a new node claims its location after joining the network. On joining the network when a new node claims its location, then agent applies our proposed protocol and tries to verify the claimed position. If it can verify the position then adds the claimed location in its list of deployed sensor nodes. Otherwise, it rechecks the claimed location with the setup server to get to know whether setup server has deployed any node in that location.

5.1 Basic Protocol

The agent starts this protocol by finding a triangulation Δ of the set of planar points S, where each point in S corresponds to a position of an immobile sensor node in the network. When a new node claims its position somewhere in the network, the agent identifies the triangle in which the claimed position of a new node is, and hands over the charge of the remaining part of the protocol to be executed by the sensor nodes in the positions of the vertices of the identified triangle. Agent does that by letting each sensor node representing a vertex of the identified triangle know about the claimed identity of the new node.

Remaining part of the protocol then goes as follows: Let, V_1, V_2 and V_3 be the nodes forming the triangle within which the new node claims its position.

Step 1: Each of the nodes $V_i\,(i = 1, 2, 3)$, generates a random number R_i and attaches with it its own identity yielding a messages M_i where $M_i = R_i V_i$.

Step 2: Each of the nodes $V_i\,(i = 1, 2, 3)$, then sends M_i to the new node X that claims its identity to be in the triangle.

Step 3: Node X receives three messages $M_i\,(i = 1, 2, 3)$, one from each of the nodes and constructs $M_i' = R_i X$ and then sends it back to the node $V_i\,(i = 1, 2, 3)$. Messages M_i are sent using radio frequency channel, and messages M_i' are sent using sound channel.

Step 4: Each node V_i $(i = 1, 2, 3)$, then measures the elapsed time t_{ii} between the delivery of the message and the receipt of the corresponding message M_i' and reports back to the software agent by sending t_{ii}.

So, if we assume that the processing time in each node is almost close to zero, then,

$$t_{ii} = d(V_i, X)(1/c + 1/s) \tag{1}$$

where $d(X, Y)$ denotes the euclidian distance between the nodes X and Y, and c and s are the distances traversed in unit time in radio frequency and ultra sound channel. The notion of using two channels for computing t_{ii} values has been borrowed from [6]. The reason why in Step 1, each node V_i $(i = 1, 2, 3)$, generates a random number to compute the message M_i is straight forward. This is required to ensure that all the nodes V_1, V_2 and V_3 and also the newly claimed node X is actively participating in the protocol and thus is facilitating in correctly computing t_{ii} values. Otherwise, it could have been possible for a malicious node to defeat the protocol by replying early (if it knows in a priori these R_i's).

If the new claimed node X is a honest one, which claims its position within the Inner Core of the triangle defined by the nodes V_i $(i = 1, 2, 3)$, then we could expect, for all the nodes V_i $(i = 1, 2, 3)$, $t_{ii} \leq t_{ij}$, where t_{ij} is the total time required for a message to reach V_j from V_i and then getting back a response in turn from V_j, for all $j \neq i$ and $j \in \{1, 2, 3\}$. This time also we have the same assumption of using both radio frequency and ultra sound channel and hence t_{ij} values for $i = 1, 2, 3$ and $j \neq i$ where $j \in \{1, 2, 3\}$ can be calculated as

$$t_{ij} = d(V_i, V_j)(1/c + 1/s) \tag{2}$$

It is to be mentioned here that since agent knows the locations of all the nodes V_1, V_2 and V_3 so it can easily compute the t_{ij} values at its side using (2).

So, if all the V_i's are honest, then the agent will consider the claimed node X to be an honest one if the above inequality holds for all $i = 1, 2, 3$, and for all $j \neq i$ where $j \in \{1, 2, 3\}$. If it is so, then it adds the node, location pair in its list of deployed nodes. This follows easily from (1) and (2) as for any vertex of the triangle, its distance from any point inside the Inner Core is less than or equal to its distance from the other two vertices of the triangle. This is the most ideal case which we could expect for the protocol to function properly.

Therefore, the agent is required to find a triangulation Δ of the set of planar points S such that the claimed location of the node is in the Inner Core of Δ. If no such triangulation exists then the software agent initiating the protocol consults with the setup server. If the setup server has deployed this node, then its possible for the agent to crosscheck it and in which case it allows the node to join the network otherwise rejects. Now given a point X within the convex hull of S, whether there exists any triangulation Δ of S such that X is in the Inner Core of Δ is not known to us. Even if it exists how to find out such a triangulation is also an open question. Hence as long as we do not have answers to these questions, agent can initiate the protocol by finding the *Delaunay triangulation*

of S for which we conjecture that it maximizes Inner Core for a set of planar points.

The proposed protocol exhibits the following interesting properties. It does not require a large number of messages to get exchanged in order for the protocol to work. Agent contributes to the message complexity by sending three messages while delegating the charge of the protocol to the sensor nodes forming the identified triangle. Each of the three sensor nodes forming the triangle is required to send two messages, one to the new node and another to the Agent. The newly joined node requires to send three messages. From the protocol its also clear that messages are of short length. Also the protocol does not demand from any sensor to have stored a huge amount of information excepting only its own location in the plane. Protocol does not suffer from the problem of scalability as it allows new nodes to join the network on the fly. The only restriction here is that a new node will only be allowed to join the network if it is within the convex hull of initial set of deployed nodes.

5.2 Modified Protocol to Handle Processing Delay in the New Node

In Step 4 of the protocol, we have seen that each node V_i $(i = 1, 2, 3)$ computes the value of t_{ii} and the expression for t_{ii} in (1) assumes that processing time in each node (in particular, in the new node X) is almost close to zero. But in practice, this is not the case, as the new node X needs some time to process the packets (in Step 3 of the protocol) received from each of the nodes V_i $(i = 1, 2, 3)$. To handle this situation, we modify our protocol slightly. First we introduce a parameter called *minimum processing time* (Δ_{min}) for any sensor node and we also assume that a sensor node is capable of measuring the processing time for each packet. We now modify Step 3 of the protocol as follows. New node X constructs $M_i' = R_i \Delta_i X$, where Δ_i is the time spent in processing the message from node V_i $(i = 1, 2, 3)$. So, the elapsed time t_{ii}' between the delivery of the message M_i and the receipt of the corresponding message M_i' is given by,

$$t_{ii}' = d(V_i, X)(1/c + 1/s) + \Delta_i. \tag{3}$$

Because of the modification in Step 3 of the protocol, it is now possible for each node V_i $(i = 1, 2, 3)$ to be able to compute t_{ii}' instead of computing t_{ii} and in Step 4 of the protocol it reports back to the software agent by sending the pair t_{ii}' and Δ_i. The agent then checks for all $i = 1, 2, 3$ whether $t_{ii}' - \Delta_i \leq t_{ij}$, for all $j \neq i$ and $j \in \{1, 2, 3\}$, where t_{ij} retains the same meaning as has been described in the previous section, to verify the claimed location of the new node.

This above solution works fine when $\Delta_i = \Delta_{min}$. But, the problem arises when $\Delta_i > \Delta_{min}$. The new node X may maliciously claim its message processing time to be more than Δ_{min}, while the original processing time is very close to Δ_{min}. Doing this helps the node X to establish its location to be inside the Inner Core of the triangle, while actually being located somewhere outside the Inner Core. This is because, by choosing Δ_i's large enough node X can force $t_{ii}' - \Delta_i \leq t_{ij}$ for $j \neq i$ and $j \in \{1, 2, 3\}$ to get satisfied for all $i = 1, 2, 3$, and

thus can defeat the protocol. The problem thus for the case $\Delta_i > \Delta_{min}$ can be solved as follows. The solution works by identifying a region included in the Inner Core of the triangle such that if the claimed location of X is inside this new region then its true location has to be within the Inner Core of the triangle. As malicious node can claim its message processing time as Δ_i ($> \Delta_{min}$) in order to make $t'_{ii} - \Delta_i$ to be less than equal to t_{ij}, similarly the verifying nodes V_i ($i = 1, 2, 3$) can adjust the values of t_{ij}'s accordingly to prevent the malicious node from defeating the protocol. How adjustments to these t_{ij} values can be made and how this relates to identifying a region included in the Inner Core of the triangle is discussed below.

When node X claims Δ_i as its message processing time for node V_i ($i = 1, 2, 3$) of the triangle, then the maximum distance which a message can traverse first in radio frequency channel and then in sound channel in time $\Delta_i - \Delta_{min}$ can be calculated as,

$$s_i = \frac{(\Delta_i - \Delta_{min})}{(1/c + 1/s)}. \tag{4}$$

We now define a new region called *Region of Acceptance (ROA)* relative to a triangle T with V_i ($i = 1, 2, 3$) as vertices as follows.

Definition 2 *Region of Acceptance of a triangle (Figure 2) T with $V_i (i = 1, 2, 3)$ as vertices is defined as,*

$$ROA(T) = \{\cap_{i=1}^{3} Disk(V_i, l'_i)\} \cap T,$$

where

$$l'_i = \begin{cases} l_i - s_i, & \text{if } l_i > s_i \\ 0, & \text{otherwise,} \end{cases}$$

and $Disk(V_i, l'_i)$ is the circular region with V_i as its center and l'_i as its radius.

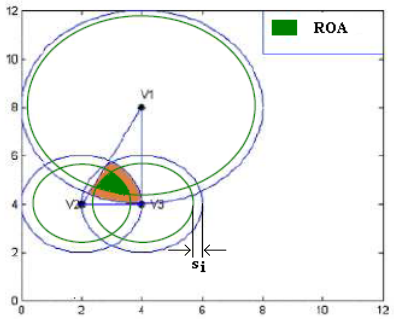

Fig. 2. Region of Acceptance of Triangle T

Its now easy to see that, if the new node X claims Δ_i for node V_i ($i = 1, 2, 3$), then $ROA(T)$ is either fully included in $IC(T)$ or $ROA(T) = \phi$. To verify the location of the new node X, the agent then does the following.

1. On having received the t'_{ii} and Δ_i pair from each of the nodes V_i $(i = 1, 2, 3)$, it first checks whether $ROA(T) \subseteq IC(T)$ or $ROA(T) = \phi$.
2. If $ROA(T) \subseteq IC(T)$, then agent computes the time t'_{ij} instead of t_{ij} for all $i = 1, 2, 3$, and $j \neq i$, $j \in \{1, 2, 3\}$ as

$$t'_{ij} = (d(V_i, V_j) - s_i)(1/c + 1/s)$$

$$\implies t'_{ij} = t_{ij} - (\Delta_i - \Delta_{min}) \text{ for all the nodes } V_i \ (i = 1, 2, 3).$$

   Then it checks to find whether $t'_{ii} - \Delta_i \leq t'_{ij}$, for all $j \neq i$ and $j \in \{1, 2, 3\}$. If it is so, then the agent considers the new node X to be an honest one. This only applies, if the claimed location of the new node is within $ROA(T)$, which the agent can easily determine by calculating the values of s_i's from the Δ_i values. If the claimed location is somewhere inside $IC(T) - ROA(T)$, then to verify the location agent has to crosscheck it with the setup server.
3. If $ROA(T) = \phi$, then the agent verifies the location by crosschecking it with the setup server.

One interesting thing to note here is that, if $\Delta_i = \Delta_{min}$ for all $i = 1, 2, 3$, then $ROA(T) = IC(T)$ and as such the agent can verify the location anywhere within the Inner Core of the triangle.

5.3 Security Analysis

Let us now consider the case when all the nodes in the set S are not honest. These dishonest nodes can help a sybil node to establish its claim of being present in the Inner Core of a triangle. In this discussion we will assume that the message processing time of a sensor node is negligible, i.e., almost close to zero. We now define *Region of Vulnerability* of a triangle T as a region such that if a malicious node is physically present in this region then it can defeat the protocol by successfully inducing a sybil node within the Inner Core of T. Definition of *Region of Vulnerability* also takes into account the fact that either no node or any number of nodes forming the triangle T can be malicious. But since every node V_i has to report back to the agent by sending t_{ii} values, so a malicious node in the position of any vertex of a triangle can't help another malicious node to launch a successful sybil attack very easily by sending erroneous t_{ii} values. An attempt to do so may help the agent to be able to isolate the malicious nodes since it is computing the t_{ij} values at its side.

Region of vulnerability for a triangle can then be calculated for the following four cases.

Case A: All the vertices of the triangle are honest. *Region of Vulnerability* of T (Figure 3) can then be calculated as

$$Rov(T) = \{\bigcap_{i=1}^{3} Disk(V_i, l_i)\} - T,$$

where the notations l_i and $Disk(V_i, l_i)$ have been described earlier.

Case B: Two of the nodes (V_i's) are honest, while the other one is not. We will see that in this scenario the protocol may fail, being unable to detect a valid

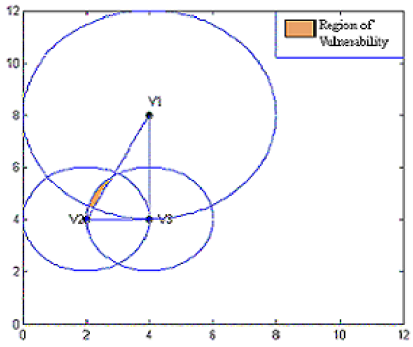

Fig. 3. Region of Vulnerability.

sybil attack. Let us consider that in triangle T with vertices V_1, V_2, and V_3 node V_1 and V_3 are honest and V_2 is dishonest. The region of Vulnerability of triangle T is shown as shaded in Figure 4. Now, a malicious node actually being present in the shaded region and claiming a sybil identity within the region defined by the triangle T, can defeat the protocol in identifying the newly claimed node as a sybil node in association with another dishonest node V_2. Node V_2 represents the distance of the new node from itself by appropriately sending t_{22} such that the agent concludes that the new node is indeed in the Inner Core of the triangle. In this case the Region of Vulnerability can be formally written as

$$Rov(T) = \bigcap_{i \in V_{honest}} Disk(V_i, l_i) - T$$

where V_{honest} represents the set of honest nodes of triangle T and such that $|V_{honest}| = 2$.

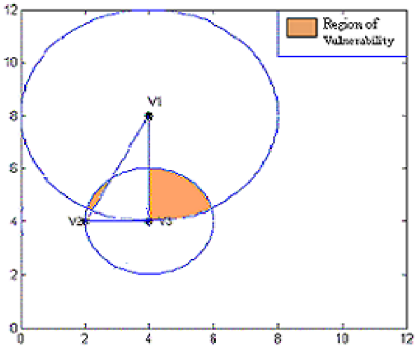

Fig. 4. Region of Vulnerability w.r.t. node V2.

Case C: One of the nodes in T is honest, and the other two are not. This case is shown in Figure 5. Node V_3 is honest, and nodes V_1 and V_2 are dishonest. The region of vulnerability of triangle T with respect to the dishonest nodes V_1

and V_2 is shown as shaded in the figure. The Region of Vulnerability in this case can be formally written as

$$Rov(T) = Disk(V_i, l_i) - T,$$

where V_i ($i \in \{1, 2, 3\}$) is the only honest node of T.

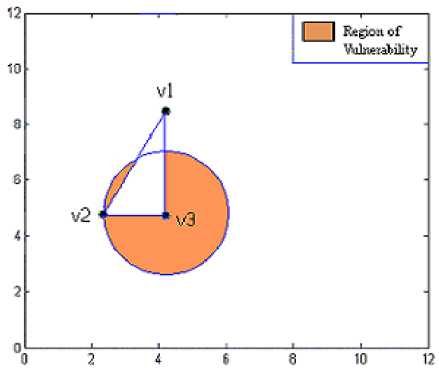

Fig. 5. Region of Vulnerability w.r.t. nodes V1 and V2.

Case D: All the three nodes of the triangle are dishonest. In this case wherever the malicious node be physically present it can always launch a successful sybil attack and thus we have $Rov(T) = ConvH(S)$.

Region of Vulnerability for a triangulation Δ of a set of nodes S can be equivalently defined as the union of the region of vulnerabilities of its constituent triangles, i.e.,

$$Rov(\Delta) = \bigcup_{T \in \Delta} Rov(T).$$

Having defined the $Rov(\Delta)$ for any triangulation Δ of the set of nodes S, we will now calculate the probability that a sybil node will remain undetected by our protocol. This probability is simply the area of the region, the physical presence of a malicious node in which can help in successfully launching a sybil attack, divided by the area of the region inside the convex hull of the set of nodes S. This region, physical presence of a malicious node in which can help in successfully launching a sybil attack is essentially the intersection of $Rov(\Delta)$'s for all such Δ. So the probability that sybil attack remains undetected under the proposed protocol is given by,

$$\frac{\bigcap_{\Delta \in \mathcal{C}} Rov(\Delta)}{ConvH(S)},$$

where $\mathcal{C}$ denotes the class of all triangulations of the set of planar points S. Since we don't have the answers to some of the questions related to Inner Core and as we have planned to use Delaunay triangulation $D \in \mathcal{C}$ for the time being, so the probability that sybil attack remains undetected is given by,

$$\frac{Rov(D)}{ConvH(S)}.$$

6 Conclusion

In this paper, we have proposed a protocol which can prevent simultaneous sybil attack in a sensor network and our protocol is meant for those sensor networks for which planned deployment of the nodes is required. The protocol works by securely verifying whether the physical position of the new node is within a region. One interesting property of the protocol is that its scalable as it allows new nodes to join the network on the fly. Our protocol takes the help of a new functional for planar triangulation called Inner Core which we have defined in this work. We have left three questions as open regarding this new functional and one in the form of a conjecture. Two open question that are of particular interest for the sake of the protocol are: (i) Is it possible to derive a characterization such that given a point X and a set of points S = V_i : i = 1, 2,.... ,n; we will be able to answer whether a triangulation Δ of the convex hull of S exists such that X is in Inner Core of Δ, and (ii) If yes, how can we find a polynomial time algorithm to find such a triangulation. Being able to answer these two questions will have significant impact on the performance of the protocol as it will help in minimizing the number of queries that an agent has to make to crosscheck the location of a node with the setup server.

References

1. R. Needham and M. Schroeder. "Using encryption for authentication in large networks of computers", In *Communications of the ACM*, pages 993 - 999, 1978.
2. D. Denning and G. Sacco. "Timestamps in key distribution protocols". In *Communications of the ACM*, 25:533-536,1982.
3. J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. "Providing Robust and Ubiquitious Security Support for Mobile Ad-Hoc Networks". In *Proceedings of International conference on Network Protocols*, 2001.
4. J. R. Douceur. "The Sybil attack". In *Proceedings of IPTPS02 Workshop*, Cambridge, March 2002.
5. W. Wang, Y. Zhu, and B. Li. "Self-Managed Heterogeneous Certification in Mobile Ad Hoc Networks". In *Proceedings of IEEE Vehicular Technology Conference*, 2003.
6. N. Sastry, U. Shankar and D. Wagner. "Secure Verification of Location Claims". In *Proceedings of the ACM workshop on Wireless security*, pages 1 - 10, San Diego, CA, USA, 2003.
7. C. Karlof and D. Wagner. "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures". In *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.
8. J. Newsome, E. Shi, D. Song, and A. Perrig. "The Sybil Attack in Sensor Networks : Analysis and Defenses". In *Proceedings of Third International Symposium on Information Processing in Sensor Networks*, April 2004.
9. B. Parno, A. Perrig, and V. Gligor. "Distributed Detection of Node Replication Attacks in Sensor Networks". In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, May 8-11, 2005.
10. Q. Zhang, P. Wang, D. S. Reeves, P. Ning. "Defending Sybil Attacks in Sensor Networks". In *Proceedings of the International Workshop on Security in Distributed Computing Systems*, June 2005.