A Novel Secret Location Sharing Scheme for the Wireless Sensor Network

Monjul Saikia, Non-member

ABSTRACT

The wireless sensor network is a collection of sensor nodes that operate collectively to gather sensitive data from a target area. During the process of data collection, the location of sensor nodes from which data is originated is important when making decisions at the base station. Location, i.e., the coordinates of a sensor node, need to be shared among other nodes in various circumstances such as the key distribution phase, during the routing of packets, etc. Since the location of every sensor node must be kept secret, a scheme that facilitates secure location sharing among sensor nodes is crucial. In this paper, we propose a novel, secure, and robust mechanism for location sharing using a two-threshold scheme. The implementation process of the proposed model is shown here along with results and analysis.

Keywords: Wireless Sensor Network, Privacy and Security, Cryptography, Secret Sharing, Affine Combination, Geo Positioning System, GPS

1. INTRODUCTION

The sensor network has computing and communication capabilities for monitoring purposes and collecting environmental data. With the help of gathered information, a decision can be taken on a specific region from which the data was initiated. In the past, sensors were connected with the aid of wires and prior knowledge of each sensor node's location. However, today's technology facilitates wire-free communication systems. The combination of GPS (geo positioning system) with sensor nodes can be used in many applications such as traffic management systems, air pollution monitoring systems, etc. For example, fixing a sensor node with a GPS module to vehicles can give real-time road traffic information. Through making use of GPS-enabled mobile devices, wireless networking, and spatial database management systems, many location-based service providers currently provide customers with services relevant to their locations. Online social networks have been integrated with location-based services, where user-generated, geo-tagged information is exchanged by individuals who are part of a social network. These location-based social networking systems with location sharing services rely on a central server from which all users in the system receive location information [1]. In some current systems, the central server also knows the approximate location of the user [2]. Some systems require the exchange of multiple messages, not only between the user and central server but also directly between the user and their friends [3].

In the past, such applications have been discussed by researchers. Misra *et al.* [4] proposed a location tracking system for a vehicle on a real-time roaming path. In a real-time system like GPS, the key emphasis is on characterizing the intercommunication of various technologies for position gathering, including the security aspects. For instance, a system that can be mounted on a vehicle and communicate with GPS satellites should be considered to be a GPS module. In the event of server communication failure, this module should be able to save a few records. The GPS module should also save data from other sensors. Tseng *et al.* [5] present an application for location tracking by mobile agents and implements a data fusion strategy. In their approach, upon identifying a new object, a mobile agent will be initiated to monitor its roaming route. The agent is mobile because the sensor closest to the object is chosen to stay. The agent may invite nearby slave sensors to position the object cooperatively and inhibit other irrelevant or distant sensors from tracking it. In this way, contact and detection overheads are significantly reduced.

Savarese *et al.* [6] stated that wireless sensing nodes rely heavily on the ability to establish position information, presenting an algorithm for range measurement between pairs of nodes with the help of sparsely located anchor nodes. Via assumptions, tests, and iterative refinements, clusters of nodes surrounding the anchor nodes collectively create confident location estimates. These positions are propagated to more distant nodes once created, enabling the entire network to construct an accurate map of itself.

Rabaey *et al.* [7] stated the importance of the

Manuscript received on January 21, 2021 ; revised on April 5, 2021 ; accepted on July 23, 2021. This paper was recommended by Associate Editor Kampol Woradit.

The author is with the Computer Science and Engineering Department, North Eastern Regional Institute of Science and Technology, India.

Corresponding author: msk@nerist.ac.in

Digital Object Identifier 10.37936/ecti-eec.2021193.244940

sensor nodes' position since most of these data have the common characteristic of being helpful only when considered in the context of where the data were measured. Therefore, most sensor data will be stamped with position information, with the primary objective of establishing a location subsystem capable of providing position estimates with consistent error margins for individual nodes within a network such as Picoradio.

Patwari *et al.* [8] used the received signal strength (RSS) or time of arrival (TOA) of signals for sensor location estimation. However, the method requires extra hardware since the purpose and accuracy are not always granted. However, the author claims that the method is an accurate and low-cost localization system.

Mao *et al.* [9] discussed various measurement techniques for localization such as angle-of-arrival (AOA), distance-related, and RSS profiling.

In the aforementioned studies, any security challenges that may arise concerning knowledge of the location by an attacker were not considered. However, a few researchers have attempted to provide secure location sharing.

In this regard, Perrig *et al.* [10] discussed various security challenges, while Zhang *et al.* [11] discussed the weakness of localization algorithms and proposed a model for a Secure Localization Scheme (SLS) with some manipulations over the signals' TOA to protect localization from adversarial attacks. Schlegel *et al.* [1] introduced a new encryption principle called Order-Retrievable Encryption (ORE), which is a new cryptographic protocol for social networking networks that implement Privacy Preserving Location Sharing Services (PPLSS). Ozturk *et al.* [12] proposed a routing algorithm for source-location privacy in an energy-constrained sensor network. Location-privacy issues, as well as the amount of energy consumption in the sensor network, were discussed. Motivated by the results, they suggested a versatile routing solution, referred to as Phantom Routing, which protects the source's location.

Shi *et al.* [13] discussed security issues in designing a secure sensor network, while Li *et al.* [14] proposed a robust statistical method for securing wireless localization in sensor networks. Teymorian *et al.* [15] proposed an application for an underwater wireless sensor network using localization methods.

In this paper, we use the concept of projective geometry, details of which can be found in [16], which also discusses the foundation of projective geometry application and analysis of invariant geometric properties in projective transformation. The results imply that, in contrast to elementary Euclidean geometry, projective geometry has a different setting, projective space, and a selective collection of basic geometric principles.

1.1 Motivation

Various methods of localization techniques have been proposed in the previous literature. Some of these may give higher accuracy in estimating a sensor node's physical location while others are lacking in this regard. All these methods require extra hardware such as sensing the received signal strength, the angle of reception, time of arrival, etc. It is difficult to estimate a sensor node's location when it resides in a multi-hop distance and the security thread seems high. From a security point of view, estimating the unauthorized user's location may lead to the system being in danger. Moreover, at the same time, accurate location information is desirable for real-life application purposes and hence secrecy is the primary motive of this work. Therefore, a secure, robust mechanism for accurate location sharing among nodes and base station is indeed a vital requirement.

1.2 Organization of the Paper

A basic introduction to the wireless sensor network and the importance of location information are provided in Section 1, together with research on localization techniques. The 2-threshold scheme of secret sharing is discussed in Section 2. In Section 3, we elaborate on our proposed model for secret location sharing in the wireless sensor network. In Section 4, we derive the mathematical equations for the required computations. The results and analysis are presented in Section 5, while Section 6 concludes the paper.

1.3 Contribution to the paper

The objective of this paper is to secretly deliver/transmit the location of sensor nodes over a network. The 2-threshold secret sharing scheme method is used to design a secure and robust mechanism for location sharing. The algorithm uses a secret key, pre-loaded in sensor nodes, and finds a secret share, which is then transmitted over the network. At the receiver end, the same is decoded using shared secret keys. Here, we propose secure location sharing using Blakley's 2-threshold Secret Sharing scheme [17] with minimal computation cost. Blakley's t-threshold secret sharing scheme is based on projective geometry concepts.

2. THRESHOLD SCHEME FOR SECRET SHARING

Let us assume that there is a vault in a bank that must be opened every day giving rise to a secret sharing situation. Several senior tellers are hired by the bank, who are sufficiently trusted to assist in opening the vault but do not trust them to own the combination of the vault itself. The threshold scheme is a protocol for the secret sharing of information over an insecure channel. The protocol shares secret X

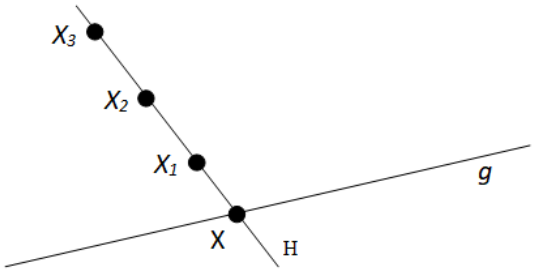


Fig. 1: *2-threshold scheme.*

among a set of participants so that participants in the specified subset can recover X by pooling their information. In contrast, participants in the other subsets are unable to recover any information on X.

The most primitive form of a secret sharing scheme is the threshold scheme, introduced by Blakely [17] and Shamir [18]. Blakely used a structure from projective geometry, while Shamir borrowed results from polynomial interpolation.

Participants denoted by A_i are interested in recreating the secret in any secret sharing scheme. There is a distributor D called the 'Dealer', who distributes information about the secret. The pieces of information are called 'Shares' or 'Shadows'.

X_i denotes a share given to participant A_i. A group of participants collecting the pieces of share together is called a 'Constellation'.

If a constellation of users can reconstruct the secret, then it is called a legal constellation, otherwise, it is an illegal constellation.

Definition 1: A t-threshold scheme is a secret sharing scheme with the following properties.

- At least t participants are required to reconstruct the secret.
- Any constellation with $t-1$ or fewer participants is illegal.

2.1 2-Threshold Scheme

A projective plane $P = PG(2, q)$ is used, where PG stands for projective geometry, the '2' is the dimension, while q is the order of the plane and $g \in P$ is a fixed public line. Since the dimension is two, we are working in a plane with points and lines.

Point X on public line g is chosen as the secret. Then we choose a hyperplane H through X not containing g (here H is a line). Then we choose a set S of points on H that are in a general position containing X and distribute the points X_i as shares to participants, as shown in Fig. 1. It can be clearly observed that $H \cap g = X$ gives the secret and a legal constellation requires at least two shares to construct H first, then $H \cap g = X$. Hence, the name 2-threshold scheme.

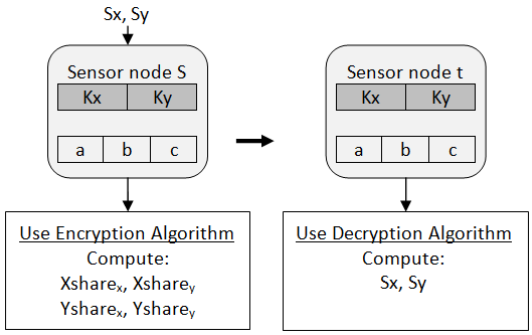


Fig. 2: *Secret location sharing model.*

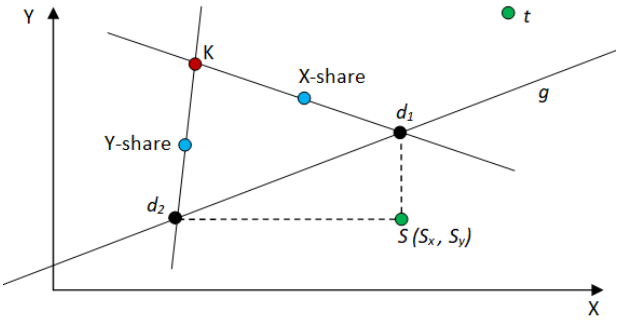


Fig. 3: *Secret location sharing: geometric representation.*

3. SECRET LOCATION SHARING: THE PROPOSED MODEL

Each sensor node is assumed to be equipped with a GPS module to detect its own location (i.e., longitude and latitude coordinates). Let S be the source node that wants to send its location (S_x, S_y) to a destination node t. Assuming that a public line g is shared among all sensor nodes S_i, let K be a shared secret key among S and t distributed prior to deployment. Then S calculates two shares; one for the X-coordinate and one for the Y-coordinate, called X_{share} and Y_{share}, as shown in Fig. 2. This geometric representation is shown in Fig. 3.

X_{share}: Let d_1 be the point on g whose X-coordinate is the same as the X-coordinate of S (i.e. S_x). S then computes a random point on line kd_1, called X_{share}.

Y_{share}: Let d_2 be the point on g whose Y-coordinate is the same as the Y-coordinate of S (i.e. S_y). S then computes a random point on line kd_2, called Y_{share}.

S sends both shares to t, while t at the other end computes d_1 and d_2 with the help of public line g, the intersection of kd_1 and kd_2, respectively. The X-coordinate at d_1 and Y-coordinate at d_2 are the actual coordinates of the transmitted S.

4. MATHEMATICAL COMPUTATIONS

The notations used are given in Table 1.

Table 1: *Notations.*

Notation	Meaning
S_i	Sensor node i
(S_x, S_y)	Sensor node location with x and y coordinates
(K_x, K_y)	Secret key
a, b, c	the coefficient for the pubic line g
X_{share}	Encrypted x-coordinate of a sensor location
Y_{share}	Encrypted y-coordinate of a sensor location
X_d	Decoded x-coordinate
Y_d	Decoded y-coordinate
χ_x	x-ccodinate of X_{share}
χ_y	y-ccodinate of X_{share}
ψ_x	x-ccodinate of Y_{share}
ψ_x	x-ccodinate of Y_{share}

A line can be represented by a linear equation with two variables; x and y and takes the form of Eq. (1).

$$ax + by + c = 0. \tag{1}$$

Therefore, the public line g, $ax + by + c = 0$ require three coefficients; a, b, and c with a large random real number to be stored at each sensor node.

4.1 Computation of X_{share}

The X-coordinate at d_1 is S_x, we need to compute Y-coordinate at d_1 by solving Eq. (1) for y to get the point $d_1\left(S_x, -\left(aS_x + c\right)/b\right)$.

Let the secret key K be a point with the coordinate (K_x, K_y). Therefore, line Kd_1 can be expressed as Eq. (2).

$$y - K_y = \frac{-\frac{aS_x + c}{b} - K_y}{S_x - K_x}(x - K_x) \tag{2}$$

Let n be a random number. For $S_x \neq K_x$, we compute a random point on line Kd_1 as follows:

$$x = (S_x - K_x)n + K_x \tag{3}$$
$$y = m(x - K_x) + K_y \tag{4}$$

where $m = \left(-\frac{aS_x+c}{b} - K_y\right)/\left(S_x - K_x\right)$ is the slope of line Kd_1 and n is a random number, provided by the pseudo random number generator.

Point (x, y) given in Eqs. (3) and (4) is the required X_{share}.

4.2 Computation of Y_{share}

The Y-coordinate of d_2 is S_y, we need to compute X-coordinate at d_2 by solving Eq. (1) for x to get the point $d_2\left(-\left(bS_y + c\right)/a,\ S_y\right)$.

Therefore, the coordinate of point d_2 is $(-(bS_y+c)/a,\ S_y)$ and the slope of Kd_2 is $m = (S_y - K_y)/\left(-\frac{bS_y+c}{a} - K_z\right)$. We compute a random point on line Kd_2 as the Y_{share} from Eqs. (5) and (6).

$$x = \left(-\frac{bS_y + c}{a} - K_x\right)n + K_x \tag{5}$$
$$y = m(x - K_x) + K_y \tag{6}$$

The computed (x, y) is the Y_{share}. The range of n gives the span of the X_{share} and Y_{share}.

4.3 Reconstruction of Sensor Location at Destination t

Since destination t knows the shared key K, from the received X_{share} and Y_{share}, t can compute the sensor S coordinate using the Theorem 4.1.

Theorem 1: The point at intersection P of two lines $y = ax + c$ and $y = bx + d$ is given by $P\left(\frac{d-c}{a-b}, \frac{ad-bc}{a-b}\right)$.

4.3.1 Reconstruction of X-coordinate

Let the coordinate of the X_{share} be (x_1, y_1). Therefore, the equations for the two lines can be written as

$$g : y = -\frac{a}{b}x - \frac{c}{b} \text{ and}$$
$$kd_1 : y = \frac{K_y - y_1}{K_x - x_1}x - \frac{K_x y_1 + K_y x_1 - 2x_1 y_1}{K_x - x_1}.$$

Therefore, we compute the X-coordinate of d_1 as

$$X_{d_1} = \frac{-\frac{K_x y_1 + K_y x_1 - 2x_1 y_1}{K_x - x_1} + \frac{c}{b}}{-\frac{a}{b} - \frac{K_y - y_1}{K_x - x_1}}$$
$$\implies X_{d_1} = \frac{-bx_1K_y + by_1K_x - x_1c + K_xc}{by_1 - aK_y + ax_1 - aK_x}. \tag{7}$$

4.3.2 Reconstruction of Y-coordinate

Let the coordinate of the Y_{share} be (x_2, y_2). Therefore, the equations of the two lines can be written as

$$g : y = -\frac{a}{b}x - \frac{c}{b} \text{ and}$$
$$kd_2 : y = \frac{K_y - y_2}{K_x - x_2}x - \frac{K_x y_2 + K_y x_2 - 2x_2 y_2}{K_x - x_2}.$$

Therefore, we compute the Y-coordinate of d_2 as

$$Y_{d_2} = \frac{\left(-\frac{a}{b}\right)\left(-\frac{K_x y_2 + K_y x_2 - 2x_2 y_2}{K_x - x_2}\right) - \frac{K_y - y_2}{K_x - x_2}\left(-\frac{c}{b}\right)}{-\frac{a}{b} - \frac{K_y - y_2}{K_x - x_2}}$$
$$\implies Y_{d_2} = \frac{ax_2K_x - ay_2K_x + cK_y - cy_2}{ax_2 - aK_x + by_2 - bK_y}. \tag{8}$$

Algorithm 1 Encryption

Require: a, b, c; $a \neq 0$ and $b \neq 0$: the coefficient of g
K_x, K_y: coordinates of secret shared key K
S_x, S_y: sensor node coordinates (plaintext)
Ensure: X_{share} (ciphertext part1)
Y_{share} (ciphertext part2)

$m \leftarrow -\dfrac{aS_x + c}{b(S_x - Kx)}$
loop: $n \leftarrow random()$
if $n == 0$ **then**
  GoTo **loop**
end if
$\chi_x \leftarrow (S_x - K_x)n + K_x$
$\chi_y \leftarrow m(\chi_x - K_x) + K_y$
$\psi_x \leftarrow (-\dfrac{bS_y + c}{c} - K_x)n + K_x$
$\psi_y \leftarrow m(\psi_x - K_x) + K_y$
return $\chi_x, \chi_y, \psi_x, \psi_y$

Algorithm 2 Decryption

Require: a, b, c: the coefficient of g
K_x, K_y: coordinates of secret shared key K
X_{share} (ciphertext part1)
Y_{share} (ciphertext part2)
Ensure: S_x, S_y sensor node coordinates (plaintext)

$x_1 \leftarrow \chi_x$
$y_1 \leftarrow \chi_y$
$x_2 \leftarrow \psi_x$
$y_2 \leftarrow \psi_y$
$S_x \leftarrow \dfrac{-bx_1K_y + by_1K_x - x_1c + K_xc}{by_1 - bK_y + ax_1 - aK_x}$
$S_y \leftarrow \dfrac{-aK_xy_2 + aK_yx_2 + cK_y - cy_2}{ax_2 - aK_x + by_2 - bK_y}$
return S_x, S_y

(X_{d_1}, Y_{d_2}) is the actual reconstructed location of sensor node S.

From the above mathematical computations, an algorithm for encryption and decryption is designed as given in Algorithms 1 and 2. The algorithm takes coefficients a, b and c of public line g and (K_x, K_y); the coordinate of the secret shared key K as input along with the sensor node's actual location (S_x, S_y). The function $random()$ generates a random number and if the value of n does not equal zero, then it computes the X_{share} and Y_{share}. This condition avoids taking the actual sensor location as X_{share} and Y_{share}. The decryption algorithm computes back the actual location of source sensor node S from the received X_{share} and Y_{share}.

5. RESULTS, DISCUSSION AND ANALYSIS

5.1 Network Setup

The sensor nodes are pre-loaded with the secret key pair (K_x, K_y) and the coefficients a, b, and c of public line g. The nodes are then deployed over the target field. When node S_i wants to send its location to S_j, it must first find the X_{share} and Y_{share} and send them to the destination node. Destination node S_j, using the secret key pair and received X_{share} and Y_{share}, evaluates the actual location of S_i.

Table 2: *Computation of X_{share} and Y_{share}.*

Sensor node Location		Secret Key K		X_{share}		Y_{share}	
S_x	S_y	K_x	K_y	X-cord	Y-cord	X-cord	Y-cord
190	20	30	80	26.3191	78.6772	29.5382	88.3119
10	20	30	80	45.9110	141.6554	23.0424	205.2363
100	200	30	80	60.6642	75.6194	−255.505	−60.797
150	50	30	80	64.0005	87.7917	69.5636	52.6098
100	100	30	80	111.043	68.4223	177.0779	106.7414

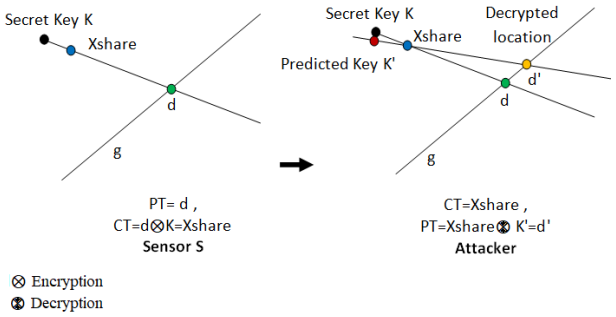


Fig. 4: *Incorrect decryption by an attacker.*

5.2 Computation of X_{share} and Y_{share}

Some of the experimental results concerning the computation of the X_{share} and Y_{share} for various sensor locations are shown in Table 2.

In the event a sensor node is moved for every new location, the values of X_{share} and Y_{share} are calculated and then transmitted to the destination node. The destination sensor node t computes the exact location of node S with the help of Eqs. (7) and (8).

5.3 Key Sensitivity

Taking the point for the secret shared key K far away from the public line and the sensor's actual location, makes it difficult to obtain the exact location of S, even if the attacker predicts a key very near to the actual secret key, as shown in Fig. 4. Some experiments on a decryption attempt by an attacker with a near equal value for the secret share are shown in Table 3. It can be observed that even though the attacker is able to use a key very near to the original secret key, he cannot evaluate the actual sensor position.

Table 3: *Reconstruction attempts at sensor location by an attacker.*

Actual Sensor Loc.		Share Key K		Predicted Key K'		Decrypted Sensor Location	
S_x	S_y	K_x	K_y	$K_x{}'$	$K_x{}'$	S_x	S_y
100	80	9999	9999	9995	9999	106.3177	9.747323
100	80	9999	9999	9996	9999	104.6825	76.5573
100	80	9999	9999	9997	9999	108.2782	79.80094
100	80	9999	9999	9998	9999	101.4888	67.22942
100	80	9999	9999	7589	1025	7227.225	9586.457

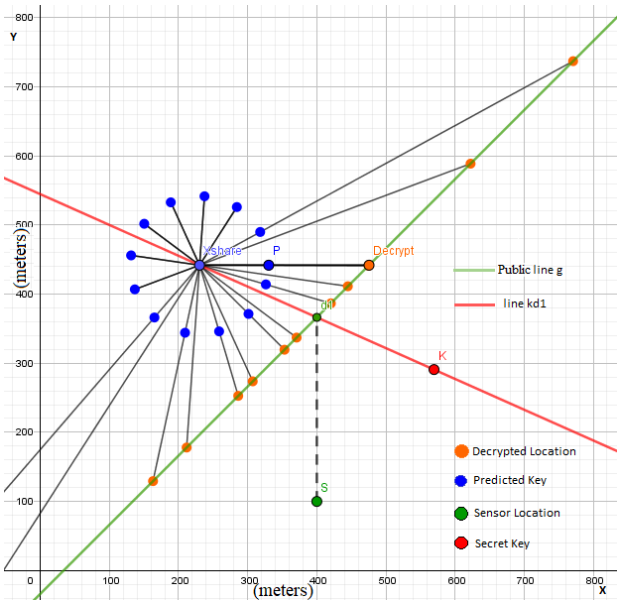


Fig. 5: *Circular key prediction by attacker.*

5.4 Circular Key prediction

An example of an attacker attempting to recompute the sensor's original location by predicting a secret key around the shared point is presented in Fig. 5 and Table 4, along with the decryption attempts to obtain the X-coordinate. The public line shared among all sensor nodes is depicted in green. The green dot S is the original location of sensor node S. It then computes point d_1 over the public line g. The computed X_{share} is then transmitted. If the attacker obtains the X_{share} and tries to decrypt the original sensor location, they will have the data for the public line g, but not the secret shared key. The attacker may then try to predict possible keys around the shared location received from sensor S (blue dots). The attacker can only find the actual location of the sensor node if they can predict a point exactly over the line through the X_{share} and secret key K. Since every time at the same sensor location the values for X_{share} and Y_{share} change, this makes it very difficult for an attacker to accurately predict the shared key.

Table 4: *Reconstruction attempts by an attacker using circular predicted keys.*

Actual Location = (400, 100)
Secret Key = (569.6192, 290.8308)
X_{share} = (230.3808, 441.9692)

Predicted Key		Decrypted X_{share}	
Px	**Py**	**Dx**	**Dy**
318.139	489.9117	770.8035	737.2035
284.411	526.1163	−209.492	−243.092
237.4545	541.7187	211.6661	178.0661
188.7661	532.8989	307.362	273.762
150.2664	501.8164	370.7272	337.1272
131.3815	456.0812	444.979	411.379
136.7351	406.8909	622.4223	588.8223
165.0164	366.2889	−1323.2	−1356.8
209.3012	344.2162	162.9719	129.3719
258.747	346.0768	286.3533	252.7533
301.2477	371.4152	353.2463	319.6463
326.3978	414.0276	420.3012	386.7012
330.3802	441.6507	474.7907	441.1907

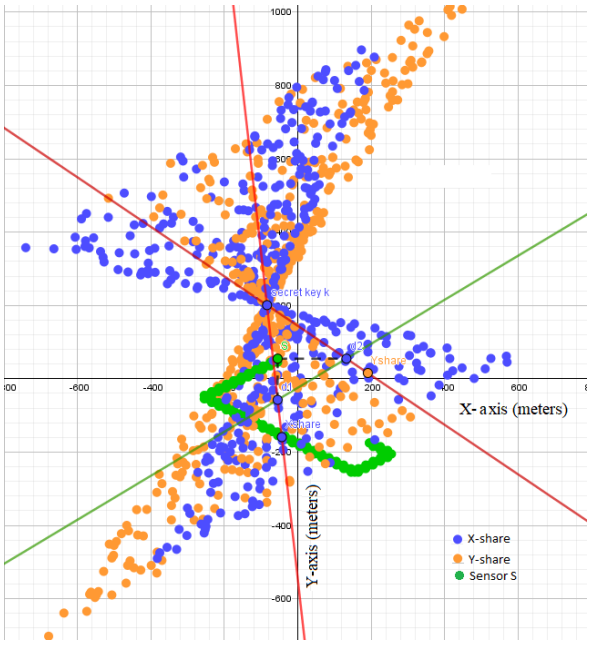


Fig. 6: *Random share of moving sensor for each location.*

5.5 Moving Sensor with Random Share

Fig. 6 shows the generation of a random location while it moves. When a sensor node moves, it shares its location by generating a secret share for a specific position. For every position the secret share is different. Therefore, it becomes harder for the attacker to predict the secret share. The blue and orange dots represent the X_{share} and Y_{share}, respectively.

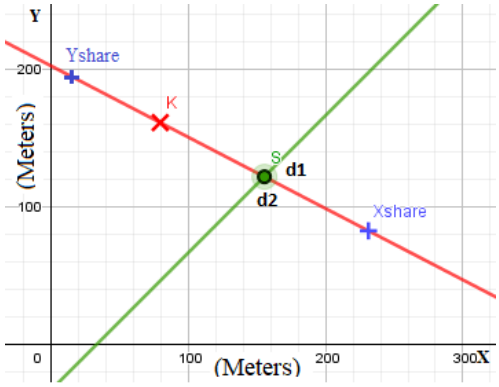


Fig. 7: *Sensor node over the public line g.*

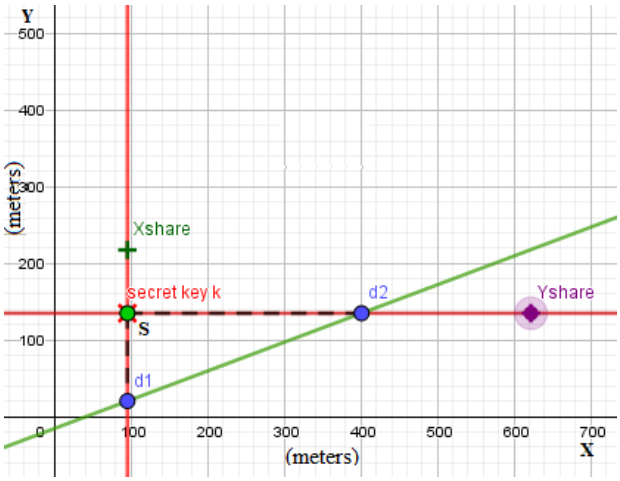


Fig. 8: *Sensor node overlap with the secret key K.*

5.6 Degenerated Cases and Solutions

Case 1: Sensor node lies over the public line g

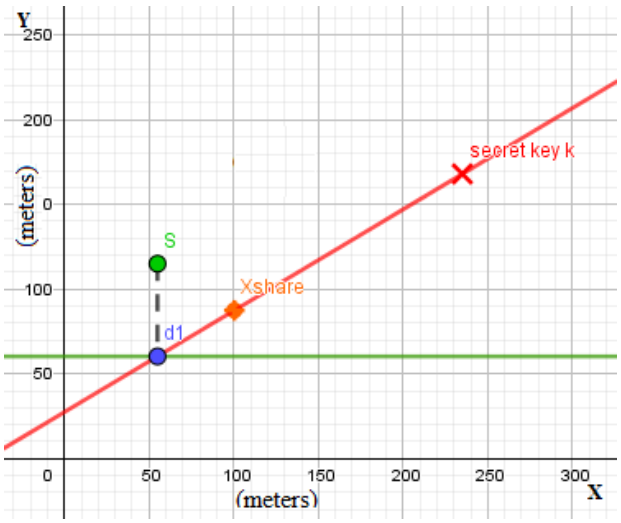
When a sensor node S lies over the public line g, the projected points d_1 and d_2 are the same as point S. Therefore, the computed X_{share} and Y_{share} are the points on the line through K and S. The method works fine in such situations despite the projected points being the same, as shown in Fig. 7.

Case 2: Sensor node overlap with the secret key K

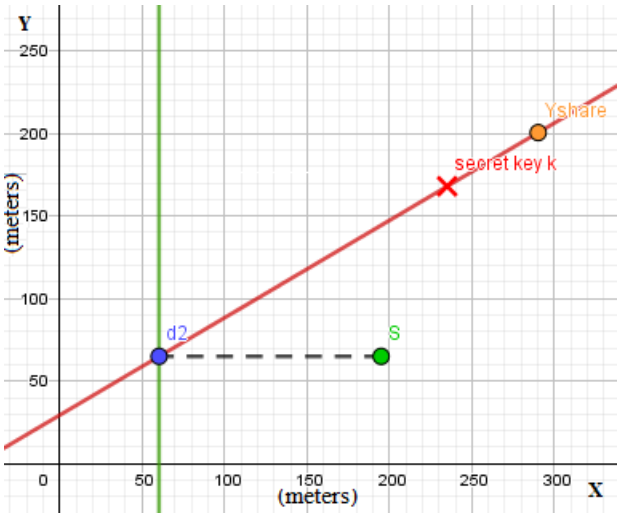
When a sensor node S overlaps with the secret key K as shown in Fig. 8, putting in an extra condition when computing the X_{share} (Algorithm 3) solves the problem. However, in practice, this does not happen because we take the point for the secret key K far away from the actual deployment area.

Case 3: The public line is parallel to any one of the axis

When the public line g is parallel to the X-axis then it is impossible to project the sensor node Y-coordinate on the line and hence we cannot evaluate the Y_{share}. Similarly, if the public line is parallel to the Y-axis then it is impossible to evaluate the X_{share}. Therefore, we restrict such orientation of the public line by putting in the condition that $a \neq 0$ and $b \neq 0$, as shown in Figs. 9(a) and 9(b).



(a) Public line parallel to the X-axis



(b) Public line parallel to the Y-axis

Fig. 9: *Public line parallel to the (a) X-axis and (b) Y-axis.*

Algorithm 3

if $(S_x == K_x)$ **then**
  $x := S_x$
  $y := \left(-\dfrac{aS_x + c}{b} - K_y\right) n + S_y$; n is a random number
else
  $x := (S_x - K_x)n + K_x$
  $y := -\dfrac{(aS_x + c + bK_y)\,(S_x - K_x)\,n}{b\,(S_x - K_x) + K_y}$
end if

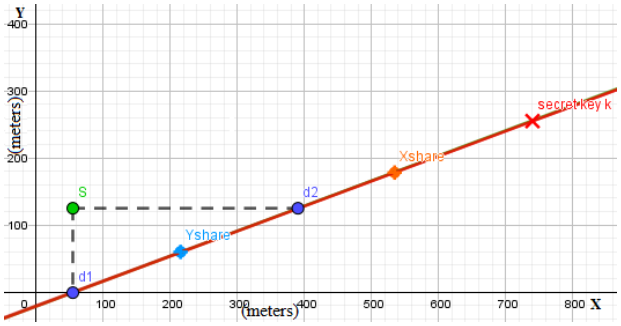


Fig. 10: *Secret Key K lies over the public line g.*

Case 4: The secret key lies over public line

The secret key K lies on the public line g and hence in such orientation both points X_{share} and Y_{share} lie on the public line. Therefore, an attacker can simply be sure that the secret key K lies somewhere on the public line, and hence easily decrypts the sensor location. Therefore, we cannot take a secret key that lies over the public line g, as shown in Fig. 10.

6. CONCLUSION

In this paper, a 2-threshold scheme is proposed for secret sharing of a sensor node's location. With the help of a GPS module attached to each sensor node, the actual coordinates are collected. A 2-threshold secret sharing scheme fits well with our location sharing scheme. We also demonstrate the various mathematical computations required in the process of encryption and decryption. Furthermore, it can be observed that the method is very robust against attack. It is very sensitive to the secret shared key, i.e., even if an attacker is able to find the secret key's approximate near value, they cannot accurately derive the source sensor node's actual location.

REFERENCES

[1] R. Schlegel, C. Chow, Q. Huang, and D. S. Wong, "Privacy-Preserving Location Sharing Services for Social Networks," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 811–825, Sep.-Oct. 2017.

[2] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new Casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd international conference on Very large data bases (VLDB '06)*, Seoul, Korea, 2006, pp. 763–774.

[3] L. Šikšnys, J. R. Thomsen, S. Šaltenis, M. L. Yiu, and O. Andersen, "A Location Privacy Aware Friend Locator," in *Advances in Spatial and Temporal Databases. SSTD 2009* (Lecture Notes in Computer Science, vol. 5644), N. Mamoulis, T. Seidl, T. B. Pedersen, K. Torp, and I. Assent, Eds. Berlin, Germany: Springer, 2009, pp. 405–410.

[4] P. Enge and P. Misra, "Special Issue on Global Positioning System," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 3–15, Jan. 1999.

[5] Y. C. Tseng, S. P. Kuo, H. W. Lee, and C. F. Huang, "Location Tracking in a Wireless Sensor Network by Mobile Agents and Its Data Fusion Strategies," in *Information Processing in Sensor Networks. IPSN 2003* (Lecture Notes in Computer Science, vol. 2634), F. Zhao and L. Guibas, Eds. Berlin, Germany: Springer, 2003, pp. 625–641.

[6] C. Savarese, J. M. Rabaey, and J. Beutel, "Location in distributed ad-hoc wireless sensor networks," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, vol. 4, 2001, pp. 2037–2040.

[7] C. Savarese, J. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," in *Proceedings of the General Track of the Annual Conference on USENIX Annual Technical Conference (ATEC '02)*, Monterey, CA, USA, 2002, pp. 317–327.

[8] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54-69, Jul. 2005.

[9] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, Jul. 2007.

[10] A. Perrig, J. Stankovic, and D. Wagner,"Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.

[11] Y. Zhang, W. Liu, and Y. Fang, "Secure localization in wireless sensor networks," in *MILCOM 2005 - 2005 IEEE Military Communications Conference*, vol. 5, 2005, pp. 3169–3175.

[12] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, Washington DC, USA, 2004, pp. 88–93.

[13] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, Dec. 2004.

[14] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, 2005, pp. 91–98.

[15] A. Y. Teymorian, W. Cheng, L. Ma, X. Cheng, X. Lu, and Z. Lu, "3D Underwater Sensor Network Localization," *IEEE Transactions on Mobile Computing*, vol. 8, no. 12, pp. 1610–1621,

Dec. 2009.

[16] A. Beutelspacher and U. Rosenbaum, *Projective Geometry: From Foundations to Applications.* Cambridge, U.K.: Cambridge University Press, 1998.

[17] G. R. Blakley, "Safeguarding cryptographic keys," in *1979 International Workshop on Managing Requirements Knowledge (MARK)*, 1979, pp. 313–318.

[18] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 1, pp. 612–613, Nov. 1979.

Monjul Saikia recieved his Bachelor of Engineering in Computer Science from Jorhat Engineering College, Jorhat Assam, India in 2005, Masters of Technology from the Department Computer Science and Engineering, North Eastern Regional Institute of Science and Technology (NERIST), India in 2011, and PhD in the Department of Electronics and Communication Engineering, NERIST in the field of Wireless Sensor Networks. He has been serving as Assistant Professor in the Department of Computer Science and Engineering, NERIST a Deemed to be University under the Govt. of India, in Arunachal Pradesh, India since July 2007. His major research interests include Information Security, Cryptography, Signal Processing, Sensor Network etc. He is a member of professional societies like IEEE, CSI (India), IEI (India) and ISTE (India).