Security for Efficient Data Transfer by Hiding the Location of Source

RINTU JOHNSON 1, Dr.K.SRIDAR 2
1 PG Student, 2 Assistant Professor
1 Department of CSE, Veerammal Engineering College
2 Department of CSE, Veerammal Engineering College

Abstract—Wireless sensor networks (WSNs) are composed of a large number of smart devices that collaborate with each other to perform various tasks. Due to the developments in sensor technology, circuit engineering, and information techniques, WSNs have been widely used in many fields, including wild habitat monitoring, target tracing and military surveillance. The information passed has to be secured from external attacks and the packets have to be delivered properly. An anonymity cloud is to hide the location of the source node so that the adversary cannot distinguish the node. The sensor nodes contain large number of nodes that can communicate from one to another. Initially a message will be broken into multiple shares, each and every share will be provided by an identifier. K-means clustering algorithm is used to group the nodes. There are a group of real shares and the fake shares that will be passed based on some time delay to confuse the adversary. The cluster methodology is used to overcome the overhead caused by the fake shares. At the time a cloud will be formed to protect the source node. The phantom routing will help to easily route the packet throughout the network. A node may receive fake shares as well as real shares, once it receives a real share that particular node will be converted to fake node to confuse the adversary. A key will be provided to secure the sink transmission by some cryptographic method; this will give much more protection for the message transmitted. The message will be reconstructed only when a node obtains 't' shares.

Keywords—Source-location privacy protection, data confidentiality, anonymity cloud, message sharing, wireless sensor networks

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of quite a number of tiny, green and smart nodes. In general, these nodes can automatically compose a connected network, though each node can directly communicate with only several neighbors which can greatly save the energy of the nodes. Based on the properties of economy and flexibility, WSNs have been extensively employed in military surveillance, industrial control, disaster management, forest fire detection and traffic monitoring. In general, the nodes that find the target are defined as the source nodes and they are responsible for reporting the target-related information to the sink node. A huge challenge is how to
Hotspots can be formed for different reasons, e.g., when the monitored wild animal in a multi-hop manner without leaking the location privacy of the target node in a multi-hop manner without leaking the location privacy of the monitored target.

In real life, the target-related information is hidden in packages before being transmitted. It is likely that the adversary can obtain the location, status and some other description information of the target if he can decrypt the packages. Many schemes have been used to protect the security of data transmission between the nodes in the network. Therefore, it is extremely difficult for the adversary to decrypt the transmitted packages. Except for content privacy, the adversary may also attempt to collect the contexture privacy of the network and find the target by analyzing the network traffic. In general, there is always a hot-spot around the source node and the adversary can find the source node by performing hot-spot attack.

Recently, many schemes have been designed in the literature to protect the source-location privacy in distributed sensor networks. Due to the developments in sensor technology, circuit engineering, and information techniques, WSNs have been widely used in many fields, including wild habitat monitoring, target tracing and military surveillance. In general, once the information is collected, it will be immediately delivered to the sink node in a multi-hop manner and then the information can be used by the network operator. The networks are likely to be deployed in harsh environments and all the nodes are strictly limited in resources such as energy, communication, computing, and storage capabilities.

Meanwhile, some nodes may not function properly and fail to monitor the environment or to receive and transmit packets. It is a great challenge to design data collection schemes for WSNs which need to be lightweight, reliable, and robust. WSNs are vulnerable to many threats. Though numerous encryption and decryption techniques have been used to protect the security of data and networks, some contextual information-based attacks cannot be processed properly. As a novel back-tracing attack, hotspot-Locating attack is a threat to source-location privacy. In WSNs, a source-location is defined as the location of the node that keeps the target monitored and source-location privacy is the confidentiality of the source node’s location. Moreover, a set of neighboring source nodes form a hotspot that generates a large data transmission amount causing an obvious inconsistency in the network traffic. Once a hotspot is located, a set of source nodes can be found.

In general, hotspots can be formed for different reasons, e.g., when the monitored wild animals have high density or spend some time in one area due to the availability of food, water, shadow, shelter, etc. The nodes in WSNs are wirelessly linked and hence the adversary can detect the radio distribution through a spectrum analyzer. Considering that a sensor node keeps silent for most of time until targets are detected, the adversary can easily trace back to the source nodes by analyzing the radio behaviors of the nodes in the networks. At last, he can locate the surrounding targets monitored by the source nodes with a high probability. It can be observed that hotspot-Locating attack is easy to implement with a low cost and it is a huge threat to WSNs. As an example, a wildlife protection organization deploys a WSN to monitor wild pandas and the collected information is periodically reported to the sink node for further analysis. In this scenario, the hunters can locate the pandas through hotspot Locating attack and apparently, this is a great threat to the pandas. Consequently, it is very meaningful to design source-location privacy protection schemes.

For example, we use pandas to represent the monitored target, though the target can be any monitored object in real WSNs. In source-location privacy protection schemes, two adversary models, i.e., global adversary and local adversary are widely used. Global adversaries are assumed to be capable of monitoring the whole network and know all the radio transmissions in the data link layer. This model is impractical for extremely large WSNs. Moreover, if the adversaries can monitor the whole WSN, i.e., deploying a parasitic sensor network (PSN) with a similar size to that of the WSN, they can directly locate the targets (e.g., the pandas) by the PSN. How to stop the adversaries from locating targets directly by a PSN is very challenging.

As a consequence, the local adversary model which assumes that the adversaries have limited overhearing capability and a parasitic node can only monitor the local area at a given time. In
general, the overhearing range $R_o$ of the parasitic nodes is similar to the communication range of the sensor nodes $R_c$ and for convenience, set $R_o$ equals to $R_c$. In the common back-tracing attack, once a parasitic node monitors a package transmission made from node $A$, it moves to $A$ and waits until another package is sent from node $B$. Then, it moves to $B$ and waits to find another package transmission. The parasitic node repeats the above process until it locates the source node. Random routing algorithms can be employed to defend this attack. However, hotspot-Locating attack is much stronger and the adversary uses traffic inconsistency caused by hotspot areas to locate pandas by analyzing the data collected by parasitic nodes. Though random routing algorithms can change the routing paths, they cannot hide the traffic inconsistency between hotspot areas and normal areas. Consequently, it is severe to provide novel source-location privacy protection approaches.

II. RELATED WORKS

A. Source-Location Privacy Protection

An adversary can deploy parasitic sensor nodes into wireless sensor networks to collect radio traffic distributions and trace back messages to their source nodes. Then, he will be able to find the source node by finding the radio signals. A Source-location privacy Protection scheme based on Anonymity Cloud (SPAC) is used to protect the privacy. A light-weight $(t,n)$–be the message and a light weight message sharing method is used to send the message, the original message are been shared to a set of message shares which are shorter in length and have minimal energy in processing and delivering. Based on the minimal shares an anonymity cloud is created to confuse the adversary and protect the privacy of the node. The anonymity cloud is a collection of nodes that are indistinguishable from each other that have same radio actions.

B. Active Trust: Secure and Trustable Routing

It is to avoid the black hole. The active trust model will improve the data route security and avoid the black holes by finding a number of paths to transmit the data from one to the other so that it can easily identify the source and the path to be transmitted. The major aspect of the method is, it fully uses the energy in the non-hotspot region to identify the route to be transmitted and will be able to achieve the desired security and energy. The detected routes are given to the active trust scheme so that it can stop the occurrence of the black hole activities in the network. Many theoretical analysis and experimental analysis have identified that the mechanism is more effective to remove the involvement of black holes. The active trust model will enhance the improvement in data route success probability and the ability to protect the network against black hole attack.

III. METHODS

A. Network model

Consider a huge 2-D network composed of a large number of homogeneous sensor nodes. Each node in the network is assumed to be able to locate itself in proper manner. They can further get their neighbors’ locations easily based on simple beacon communications. Further assume that each node is capable of computing, communication, and storage to properly execute the instructions. , a sink node acts as a bridge between a network and the network operator and it is much more powerful than the common nodes. Therefore, assume that the sink nodes in our network have sufficient resources in terms of computing, storage, and data transmission.

The deployed nodes in the network employ the $k$ -Nearest neighbors tracing approach to monitor the targets. Each node in the network will follow a sleeping schedule and will be silent when
there is no target detected. However, if a node detects a target in its region of responsibility, it needs to keep active until the target moves out of the region. In general, a target is simultaneously detected by a set of nodes and assumes that these nodes can locate the target accurately in a cooperative manner. At last, the information of the target is sent to the sink nodes in time.

The target level is the foundation of the whole network. Individuals or organizations need to deploy WSNs to collect the information about the targets such as the locations and physical condition. All the parasitic nodes compose the PSN level.

![Fig.1. Levels of network](image)

**B. Enhanced Hotspot-Locating attack**

Each parasitic node can monitor radio signals locally and locate the sender of the messages. However, they cannot locate the receiver of the packages, because any node in the transmission range can be the receiver. The parasitic nodes can communicate with each other by wireless links and they can share the collected data in time. In this way, a set of parasitic nodes in a near area can form a more powerful organization and the monitor radius greatly enlarged compared with a single node.

![Fig.2. Backtracking and Trace back attack model](image)

As an example, the process of Hotspot-Locating attack is presented in Fig. 2. A parasitic node is initially deployed around the sink node and some others are distributed in the network randomly. In back tracing phase, the parasitic nodes collect traffic information including the coordinates of the nodes that sent a packet and the time of sending the packet. Then the parasitic nodes analyze the collected information and judges whether they find a hotspot or they can move to a more promising area that can lead to the hotspot. Two types of information including time correlation and packet sending rate are analyzed simultaneously to locate the hotspot.
Specifically, the adversary identifies a hotspot by using the fact that more packets are sent out by the nodes near to the hotspot compared with the nodes far away from the hotspot. Therefore, the adversary can continuously move toward the hotspot by analyzing the traffic rather than track back by a packet. As shown in Fig.3. Hotspot-Locating attack comprises of two patterns including inside back-tracing pattern and boundary back-tracing pattern. In the inside back-tracing pattern, the parasitic nodes follow the high packet sending rates of the nodes which relay the hotspot’s packets and finally reach a suspect region. A parasitic node moves from area $\mathcal{A}4$ to $\mathcal{A}5$ by employing the inside back-tracing pattern. Apparently, if a parasitic node moves out of the hotspot, the packet sending rate greatly decreases suddenly and hence it can infer the hotspot region. In the boundary back-tracing pattern, the parasitic nodes can identify the boundary easily by observing the large difference in packet sending rates between the two sides of the boundary. The parasitic nodes move on the boundary of a large packet sending rate until they reach a suspected region. In Fig. 3.4, a parasitic node moves from area $\mathcal{A}1$ to $\mathcal{A}2$ and then to $\mathcal{A}3$ by employing the boundary back-tracing pattern.

Once the adversary finds a small suspect area, then assume that the adversary collects all his resources and deploy them in this area. In the extreme case, the adversary can monitor all the nodes in suspect. It can be observed that the given attack model is much stronger than packet-based back-tracing attack.

C. Pre-Deployment Phase

To protect data privacy between a pair of nodes, first design a pair wise key negotiation algorithm based on bilinear map. Before scattering all the sensor nodes into the monitored area, each node $ni$ is assigned with a unique identifier $IDni$, a public key $PKni$ and a secret key $SKni$ which are used to negotiate session keys with its neighbors. Let $g$ be a generator of $G0$ and $e$ be a bilinear map $e: G0 \times G0 \rightarrow G1$ with the following properties:

1. Bilinearity: $\forall u,v \in G0$ and $a,b \in \mathbb{Z}_p$ 
   \[ e(ua, vb) = e(u, v)ab \]
2. Non-degeneracy: $e(g, g) \neq 1$

Let $H$ be a hash function and $H:\{0,1\}^* \rightarrow G0$. The public key of node $ni$ is calculated as follows:

\[ PKni = H(IDni) \]  \hspace{1cm} (1)

Private Key generator (PKG) randomly chooses a master key $s$ from $\mathbb{Z}_p^*$. The secret key of node $ni$ is calculated by PKG as follows:

\[ SKni = PKnis \]  \hspace{1cm} (2)
Note that, though node knows $PKni$ and $SKni$, it cannot obtain $s$ because of discrete logarithm difficulty. Similarly, node $nj$ has the public key $PKnj$ and the secret key $SKnj$. In the deployed network, a pair of neighbor nodes $ni$ and $nj$ can negotiate a session key as follows:

1. Node $ni$ selects a random number $a \in \mathbb{Z}_p^*$ and computes
   \[ Ni = PKni. \]
   Node $ni$ sends $Ni$ and $PKni$ to node $nj$.

2. Node $nj$ selects random numbers $b \in \mathbb{Z}_p^*$ and computes
   \[ Nj = PKnjb. \]
   Node $nj$ sends $Nj$ and $PKnj$ to node $ni$.

3. Node $ni$ calculates the session key as follows:
   \[ Snij = e(SKni, \ Nj \cdot PKnia) \]  \( (3) \)

4. Node $nj$ calculates the session key as follows:
   \[ Snij = e(Ni \cdot PKnib, \ SKnj) \]  \( (4) \)

Based on the properties of $e$, we can prove that
\[
Snij = e(SKni, \ Nj \cdot PKnia) \\
= e(PKnis, \ PKnjb \cdot PKnia) \\
= e(PKnis, \ PKnja+b) \\
= e(PKni, \ PKnj) s(a+b) = Snj
\]  \( (5) \)

At last, nodes $ni$ and $nj$ get a session key which can be used to securely transmit data.

D. Light-Weight Message Splitting and Sharing Scheme

First design a $(t,n)$-threshold message splitting and sharing approach based on congruence equations and then we prove its correctness and security. For message $M$ generated by a source node, we first encode it by an interleaving coder and then split it into $t$ pieces of sub-messages $x_1, x_2, \cdots, x_t$ with equal lengths. The interleaving coder is employed to destroy the semantic meanings of each single sub-message.

In Shamir’s secret sharing scheme, the shares are constructed based on $n(t-1)$ additive operations, $n(t-1)$ multiplicative operations and $n(t-1)$ exponential operations. Therefore, our scheme is of lower computation complexity compared with the classic secret sharing scheme.

Theorem 1 (Correctness): If the sink node receives at least $t$ shares constructed by congruence equations, the sink node can reconstruct message $M$.

Theorem 2 (Security): If the adversary intercepts less than $t$ shares of message $M$, he cannot reconstruct message $M$ accurately.

At first the source node will generate an original message $X$ and will map the message $X$ to a set of message segments by the help of interleaving encoder. It is to design a $(t,n)$-secret sharing scheme and hence the original message $X$ needs to be split to exactly $t$ sub-messages with equal lengths. For more convenience, assume that $X$ is denoted as a binary number. As an example, in Figure 4, we split $X$ to four sub-messages, $x_1, x_2, x_3$ and each sub-message will contain the part of the original message $X$. Moreover, each sub-message is composed of three message fragments which are obtained by extracting the original message $X$. In this way, each sub-message does not contain any semantic information. As the sub-messages does not contain any valuable information, if the adversary access the sub-message it will not be able to access the original message. However, the sink node can recover $X$ by an interleaving decoder once it can get $x_1, x_2, x_3$. 

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3
\end{array}
\]
E. Anonymity Cloud Construction Based On Message Shares

A. Spreading the anonymity cloud

To protect source-location privacy, the source node uses the shares \( s_1, s_2, \ldots, s_n \) to construct an anonymity cloud of an irregular shape in the around area. In this way, the adversary cannot locate the source node by analyzing the shape of the cloud.

The size of an anonymity cloud is defined as the number of nodes covered by the cloud and it is indirectly decided by the number of average hops \( h \) that the shares can be transmitted in the cloud. For a network with a high security requirement, the size of the anonymity cloud should be increased and on the contrary, if the network has a low-security requirement, the size of the anonymity cloud should be decreased.

Initially, source node \( T \) generates message \( M \) and 3 message shares are constructed by the message sharing scheme. Assume that \( h = 3 \) and the corresponding hop counts for the three shares are 2, 3 and 4, respectively. Then the source node sends the three shares to its neighbors randomly and the real shares are delivered in the network along with the three red paths. Meanwhile, some fake shares are also generated and broadcasted in the network to protect the real shares from being identified by the adversary. The cloud stops spreading once the hop counts of the shares decrease to 0 and the corresponding nodes, i.e., \( F_1, F_2, F_3 \), that receive the real shares are defined as the fake source nodes. Meanwhile, all the nodes that receive either real shares or fake shares with hop counts 0 are defined as candidates of fake source nodes and they may become a fake source node for the next share. By using some routing algorithm the fake source node send the share to the sink node.

B. Choosing the next hop of a real share

A new strategy to choose the next hop of a real share based on the sector based directed walk model. A real share \( s_i \) is sent from node \( A \) to node \( B \) and node \( B \) needs to select the next hop of \( s_i \). Assume that the communication radius of node \( B \) is \( R_c \) and it has 6 neighbors namely \( A, C, D, E, F, \) and \( G \). Apparently, it is unacceptable that node \( B \) sends \( s_i \) back to node \( A \) and hence the choice of the next hop must be one of the nodes in \( C, D, E, F, \) and \( G \). Considering that we do not want the real shares to be sent back and forth, we divide the whole communication range of node \( B \) into two half-circles, i.e., sector \( a \) and sector \( b \), based on line \( l \) which is perpendicular to line \( AB \) and goes through node \( B \). Then, only the nodes in Sector \( a \), i.e., \( C, D, \) and \( E \), are legal candidates of the next hop and the other nodes, i.e., \( F \) and \( G \), are illegal candidates. Node \( B \) can randomly choose a legal candidate as the next hop of share \( s_i \). The node \( B \) selects node \( C \) as the next hop of \( s_i \). However, in some cases, node \( B \) may have no legal neighbors in its communication range, in which case node \( B \) needs to choose the neighbor nearest to line \( l \) as the next hop.

C. Generating time delays for the shares
In a cloud, the fake shares contain no valuable information and they are employed to hide the real shares. When delivering fake shares in the cloud, a randomly generated time delay \( t_{fake} \sim N(\mu, \sigma^2) \) is employed by all the nodes to destroy the regular time patterns beneath the shares. If \( t_{fake} \) is too small, the time pattern cannot be destroyed thoroughly, because the shares of different messages cannot coexist in the same cloud and it is easy for the adversaries to analyze the orders of the nodes in the process of transmitting shares. If \( t_{fake} \) is too large, the freshness of message \( M \) decreases which is unacceptable for the data users. Overall, we need to set \( t_{fake} \) in a proper way to achieve a balance between security of the source node and timeliness of the data.

Now set \( \mu = 1 f / \) and \( \sigma = 1.3 f / \) which is an extension of the distribution of the real shares. In this case, we can hide the real shares in the fake shares and they are generated with the same frequency in the cloud. In fact, the radio actions of the nodes in the cloud are similar to the radio actions of the nodes in source location protection scheme.

D. Updating and merging the anonymity cloud

Though the cloud keeps stable in general, the parameter \( h_i \) of the shares need to be updated if the monitored target stays in a field for a long time and hence the cloud will be updated. When a new source node is generated near to an existing cloud, the new constructed cloud may intersect to the old one. In this case, we need to merge them to a larger anonymity cloud. Specifically, if a sensor node receives multiple fake shares from different clouds, it sends just one fake share that has the largest number of hop counts and drops the other fake shares. However, if several real shares are received by a candidate fake source node, the fake source node sends them to the sink node in order with proper time delays. Note that, the outside shares cannot be delivered to a cloud without the help of border nodes that is the candidates of fake source node.

All the shares generated from a message \( M \) are sent out by the source node at one time and hence only one anonymity cloud is constructed for message \( M \). We define the whole process of delivering a message from the source node to the sink nodes based on message sharing scheme as a message delivery round. In average, each node in the cloud needs to transmit less than one fake share in a round and some nodes may transmit at least a real share with an extremely low probability. Overall, each node in the cloud needs to transmit about one packet with the same length of real shares. As a consequence, the nodes in our scheme are much more energy-efficient than the nodes in the existing cloud-based schemes.

E. Message Delivery to the Sink Node and Reconstructing Message \( M \)

The fake source nodes send the shares to the sink node. Any existing routing algorithm can be employed to deliver the shares including both constant routing algorithms and random routing algorithms. Intuitively, random routing algorithms (e.g., Phantom routing algorithm) can be seamlessly integrated into our scheme and they can make the proposed approach perform better in protecting the source location privacy.

This is reasonable considering that random routing algorithms can further disperse the routing paths and improve the difficulty of back-tracing. In addition, to make a message indistinguishable in the routing path, we can employ the pseudonym technique.

Though several shares are sent to the sink node, the total data transmission amount does not increase significantly considering that the length of the shares is much shorter than that of message \( M \). Considering that any \( s_i \ (1 \leq i \leq n) \) can be linearly expressed by \( \{x_1, x_2, \ldots, x_t\} \), we can get \( t \) linearly independent equations with \( \{x_1, x_2, \ldots, x_t\} \) as unknown variables. Based on Theorem 1, we can solve the equations by the Gauss Elimination Method for a unique result of \( \{x_1, x_2, \ldots, x_t\} \). At last, message \( M \) can be reconstructed through an interleaving decoder based on \( \{x_1, x_2, \ldots, x_t\} \) and then the message delivery process is completed. Assume that the sink node is of sufficient power and...
hence the energy consumption of reconstructing message $M$ is ignored in this system implementation.

IV. RESULTS

To reduce the message time delay without an apparent increase in the data transmission amount an SLPP is introduced. The source node will send the message as soon as possible, while keeping them indistinguishable from the dummy messages. The overhead of dummy variables can be reduced by cluster method but there may be some time delay.

Phantom routing is a random routing technique that consists of a random walk phase that will help the packet to randomly walk for ‘$k$’ steps. Here random walk phase is transformed into the single path routing phase. The fake source node can employ any existing routing algorithm to deliver the messages to the sink node. The Phantom routing algorithm adds a random walk phase into the routing process. Apparently, this makes the routing paths diverse with each other even the source node and the sink node are constant. To defend against the local adversary, routing-based source-location privacy protection schemes are used. The model can be an efficient and secure framework for authenticated broadcast/multicast process by sensor nodes as well as for outside user authentication.

Moreover, many secure data transmission schemes were designed for cluster-based networks. Here the model use a cloud based scheme, the cloud is filled with fake messages and construct through a complex process. It uses a message sharing scheme that allows one to map an original message to several independent messages called shares. The shares can be distributed to a set of users and only certain qualified subsets of users can recover the original message. Verifiable secret sharing scheme is used to have a secured sharing. Even if the dealer is malicious there is a well defined secret that the nodes can later reconstruct.

There will be a key given to the sink node for the message to be more secured. The message will be protected by the secret key. This provides much more secured way of transmission within the network. The entire network will be protecting the source and the sink, the information about the source will be protected by the cloud based methodology and the sink along with the message will be protected by the secret key so that the adversary will find difficulty in encrypting the data and finding the information of the source.

Fig.5. screen shot (a) Nam window opens for identifying the simulation (b). Data transfer between source and destination with the help of r1 and r2
V. DISCUSSION

A. Simulation Settings

Evaluate the performance of SPAC on packet delivery layer based on ns-3 discrete event simulator (version ns-3.21). In our simulation, 6000 sensor nodes are scattered in a 4000 m × 4000 m square region. The sink node locates in the center of the network and the farthest distance between a source node and the sink node is about 40 hops. We construct the routing paths between fake source nodes and the sink node based on geographic information of the nodes. This is reasonable considering that geographic routing algorithms are of great scalability and do not strictly limit the hop counts in routing process. Therefore, they suit large WSNs very well. For convenience sake, we assume that only one panda exists in the network. The motion model of the panda is defined as follows. First, a preset moving path is generated by a cubic polynomial and the moving speed is set to be 1 m/s. Specifically, we build a coordinate system with the original point at the center of the network. Then, we randomly generate three numbers to act as the coefficients of the cubic polynomial and employ the shape of the polynomial in the coordinate system as the preset path. At last, the initial location of the monitored hotspot is randomly chosen on the path.
Through a series of simulations, it is observed that the routing-based approaches cannot provide strong protection to the source-location privacy under the enhanced Hotspot-Locating attack. The shortest path routing algorithm is the most vulnerable approach because the paths of the messages are always very similar with each other. Though this approach is greatly energy-efficient, it is useless in protecting source-location privacy. The Phantom routing algorithm adds a random walk phase into the routing process. Apparently, this makes the routing paths diverse with each other even the source node and the sink node are constant. However, it is also likely for the adversaries to trace back to the source node. When a stream is continuously transmitted to a sink node, the parasitic nodes can first trace back to the fake sources which are about \( h \) hops away from the source node. Considering that the adversary can deploy all its sources in the suspected region, he can easily locate the source node. Though some extra energy is consumed in the random walk phase compared with the shortest path routing algorithm, the Phantom algorithm is still very energy-efficient compared with the routing-based approaches, SPAC and the cloud-based scheme can provide much stronger protection on the source locations.

V. CONCLUSION AND FUTURE WORK

Anonymity based cloud security architecture is included which can protect the location of the sink node from the adversary. First design a lightweight message splitting and sharing scheme particularly for WSNs based on congruence equations. Then an anonymity cloud is developed for
keeping the source node indistinguishable. The anonymity cloud will be surrounded around the network that cannot be seen by the adversary. A particular message will be send to the sink node and this message will be splitted into multiple shares. Each and every node will be given certain identifier to recognize the node. In the model we introduce two shares; one is the real shares that contain the original information and the fake shares that contain the null information, to confuse the adversary. With a specific time delay the shares will be routing and the phantom routing methodology is used. Finally when the node receives at least t shares it will be able to reconstruct the original message. The shortest path routing algorithm has the best performance because the packages can be delivered to the sink node with the least hops. The proposed approach provides strong privacy protection with an energy-efficient manner. In addition, data confidentiality in the network and fault tolerance for the failure of sensor nodes is also greatly improved.

REFERENCES

[1] S.He , j.Chen , F. Jiang, D.Yau, G.Xing, and Y.Sun (2013), “Energy provisioning in wireless rechargeable sensor networks,” IEEE Trans. Mobile Computing., vol. 12, no. 10, pp. 1931–1942.

[2] W.Huang, M.Langberg and J.Kliwerer(2016), “Communication efficient secret sharing,” IEEE Transactions on Information Theory, vol. 62, no. 12

[3] Jun-Song Fu,Zhenjiang Zhang(2016),"K-nearest neighbors tracking in wireless sensor networks with coverage holes", Pers. Ubiquitous Computing., vol. 20, no. 3, pp. 431-446

[4] A.Liu ,Y, LiuDong and K.Ota (2017), “ActiveTrust: Secure and Trustable Routing in Wireless Sensor Networks.” IEEE Trans. Inf. Forensics Security, vol. 11, no. 9, pp. 2013-2027

[5] O.Marek and O.Urszula (2010), “The use of mathematical linguistic methods in creating secret sharing threshold algorithms,” Computers and Mathematics with Applications, vol. 60, no. 2, pp. 267-271, 2010.

[6] D.Qin,S. Yang,S. Jia , J.Zhang,J. Ma and Ding Q.(2017), “Research on Trust Sensing Based Secure Routing Mechanism for Wireless Sensor Network,” IEEE Access, vol. 5

[7] Rashid, B.Rehmani .(2016),” Applications of wireless sensor networks for urban areas”: A survey. J. Netw. Comput. Appl, 60, 192–219

[8] Smart, and P.Nigel.(2016), “Secret Sharing Schemes,” Cryptography Made Siple, pp. 403-416

[9] Wang Na, Junsong Fu. Jian Li. Bharat Bhargava, ( 2019)," Source-Location Privacy Protection based on Anonymity Cloud in Wireless Sensor Networks" IEEE transaction on information forensics and security, vol. , no. 

[10] N.Wang, and J.Zeng(2017) “All-Direction Random Routing for Source-Location Privacy Protecting against Parasitic Sensor Networks,” Sensors, vol. 17, no. 3

[11] B.Zou,Y. Liang,L. Lai, S.Shamai (Shitz)(2015) "An information theoretic approach to secret sharing", IEEE Trans. Inf. Theory, vol. 61, no. 6, pp. 3121-3136

[12] Z.Zheng, A. Liu, L. Cai, Z. Chen, andX. Shen(2016), “Energy and memory efficient clone detection in wireless sensor networks,” IEEE Trans. Mobile Comput., vol. 15, no. 5, pp. 1130–1143

[13] Radha Krishnan, B., Vijayan, V., Parameshwaran Pillai, T. and Sathish, T., 2019. Influence of surface roughness in turning process—an analysis using artificial neural network. Transactions of the Canadian Society for Mechanical Engineering, 43(4), pp.509-514.

[14] Krishnan, B.R., Ramesh, M., Giridharan, R., Sanjeevi, R. and Srinivasan, D., Design and Analysis of Modified Idler in Drag Chain Conveyor. International Journal of Mechanical Engineering and Technology, 9(1), pp.378-387.

[15] Krishnan, B.R., Vijayan, V. and Senthilkumar, G., 2018. Performance analysis of surface roughness modelling using soft computing approaches. Applied Mathematics & Information Sciences, 12(6), pp.1209-1217.
[16] KRISHNAN, B.R. and PRASATH, K.A., 2013. Six Sigma concept and DMAIC implementation. International Journal of Business, Management & Research (IJBMR), 3(2), pp.111-114.
