Polynomial Based Compromise Resilient En-Route Filtering Scheme Against False Data Attacks Networked Systems

Ms. Kanchan B. Korke   Prof. V. K. Barbudhe   Dr. A. D. Shelotkar

***Abstract*- In Cyber-Physical Networked Systems (CPNS), the adversary can insert false dimensions into the controller through compromised sensor nodes, which not only threaten the security of the system, but also consume network resources. To deal with this issue, a number of en-route filtering schemes have been designed for wireless sensor networks. However, these schemes either lack resilience to the number of compromised nodes or depend on the statically configured routes and node localization, which are not suitable for CPNS. In this paper, we propose a Polynomial-based Compromise-Resilient En-route Filtering scheme (PCREF), which can filter false injected data effectively and achieve a high resilience to the number of compromised nodes without relying on static routes and node localization. PCREF adopts polynomials instead of Message Authentication Codes (MACs) for endorsing measurement reports to achieve resilience to attacks. Each node stores two types of polynomials: authentication polynomial and check polynomial, derived from the primitive polynomial, and used for endorsing and verifying the measurement reports. Through extensive theoretical analysis and experiments, our data shows that PCREF achieves better filtering capacity and resilience to the large number of compromised nodes in comparison to the existing schemes.**

***Index Terms*- Cyber-physical networked system, data injection attack, sensor networks, and polynomial-based en-route filtering, Security.**

I. Introduction

Cyber-physical systems (CPS) are "engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical component. Cyber-physical systems (CPS) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. Just as the internet transformed how humans interact with one another, cyber-physical systems will transform how we interact with the physical world around us. Many grand challenges await in the economically vital domains of transportation, health-care, manufacturing, agriculture, energy, defense, aerospace and buildings. The design, construction and verification of cyber-physical systems pose a multitude of technical challenges that must be addressed by a cross-disciplinary community of researchers and educators.

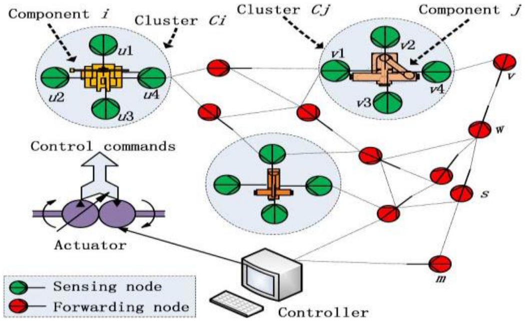


Fig .1 Numbering and spacing

improvements in digital physical arranged frameworks (CPNS) [1], [2]. CPNS, comprising of sensor hubs, actuators, controller, and remote systems, have been broadly used to screen and influence nearby and remote physical elements in the physical world [3], [4]. Ordinary CPNS spread an extensive variety of uses [5], including transportation systems, vehicular systems, systems of unmanned vehicles, and so forth. In CPNS, sensor hubs get the estimation from the physical segments, prepare the estimations and send measured information to the controller through systems. As per estimations, the controller assesses the condition of physical frameworks and sends criticism summons to actuators to control the operation of physical frameworks [6].

II. Problem Definition

In Cyber-Physical Networked Systems (CPNS), the adversary can inject false measurements into the controller through compromised sensor nodes, which not only threaten the security of the system, but also consume network resources. To deal with this issue, a number of en-route filtering schemes have been designed for wireless sensor networks. However, these schemes either lack resilience to the number of compromised nodes or depend on the statically configured routes and node localization, which are not suitable for CPNS. In this project, we propose a system called False Data Injection Attack(FDIA), which can filter false injected data effectively and achieve a high resilience to the number of compromised nodes without relying on static routes and node localization.

FDIA adopts polynomials instead of Message Authentication Codes (MACs) for endorsing measurement reports to achieve resilience to attacks. Each node stores two types of polynomials: authentication polynomial and check polynomial, derived from the primitive polynomial, and used for endorsing and verifying the measurement reports. Through extensive theoretical analysis and experiments, our data shows that FDIA achieves better filtering capacity and resilience to the large number of compromised nodes in comparison to the existing schemes.

III. Literature Survey And Review

There are numerous approaches to perform this system. Some of them are: (dynamic), Statistical, and Commutative figure based, constrained capacity based, Priority-based, Group rekeying-based, and secure ticket-based and couple of something beyond. The accompanying part of the organization will cover some of these before-specified techniques. Furthermore, LBRS has a noteworthy change over SEF, and mitigates T-edge confinement issue in SEF through area product verification key. In LBRS detecting fields are partitioned into quadrangle units, and each cell is associated with some unit keys which are determined stayed on the unit's range. Each hub stores two sorts of unit keys. In which one sort holds the keys constrained to their detecting cells to check the reports from individuals‟ cells and another sort holds the keys of some arbitrarily picked remote units. In which are extremely plausible to send their reports amid the hubs live cell. In LBRS, a forward hub affirms the got reports and sifts through false ones on the same path as SEF.

A. En-Route Filtering:

An en-route filtering mechanism’s main objective is to enhance the effectiveness of filtering and improve prevention against node compromise. Both the destination node and intermediate nodes check for the authenticity of the packet and false data is identified as early as possible in an en-route filtering schemes. Hence the number of hops the false data will travel is reduced and energy is conserved. Every intermediate node verifies the MAC computed by the previous node in the routing path and then removes that MAC from the received packet in the first phase of en-route filtering mechanism. It computes a new MAC based on its pair wise key shared with the next node to which it should forward the packet, if the verification test is passed. This new MAC attaches to the packet. Finally, it forwards the report to the next node in the route.

B. Statistical En-Route Filtering (SEF):

SEF (Statistical En-Route Filtering) is the most punctual on the way sifting strategy arranged through F. Ye. H. Lou to address the created report insertion assaults in the event of bargain hubs or introduce an on the way separating structure. In Statistical En-Route Filtering, in that is a worldwide key pool that is detached into n non-covering fragment allotment. Before organization, each hub stores a couple of verification keys subjectively favoured from one detachment of globe key pool. Hubs with keys from same parcel are considered as the same gathering. Along these lines, all hubs are separated into n bunches by means of non-covering key allotments. The SEF strategy receives T-verification, that is, the honest to goodness report must help T MACs created through T hubs from various gatherings. Each of these T hubs produces MAC with one of confirmation keys it put away. Each occasion distinguishing sensor favors the data through make a key using one of its spared keys. A report with inadequate number of MACs won’t be there forward.

At the point when the sink gets event reports, it can affirm each one the MACs passed on in the report since it has comprehensive data of the worldwide key pool. Furthermore, the false reports with wrong MACs to go amid on the way sifting resolve accordingly are taken note. At that point the SEF technique identifies alongside drops false reports as of the traded off hubs. The confirmation of the MACs is finished probabilistically. SEF which can't see in which hubs is bargain since reports are sifted on the way probabilistically however it can cease the false data implantation hit with 80 - 90 percent prospect inside 10 jumps. In SEF if a hub is mollifications the assailant can obtain the keys for numeral of traded off hubs since more than one hub gathers keys from conventional key pool.

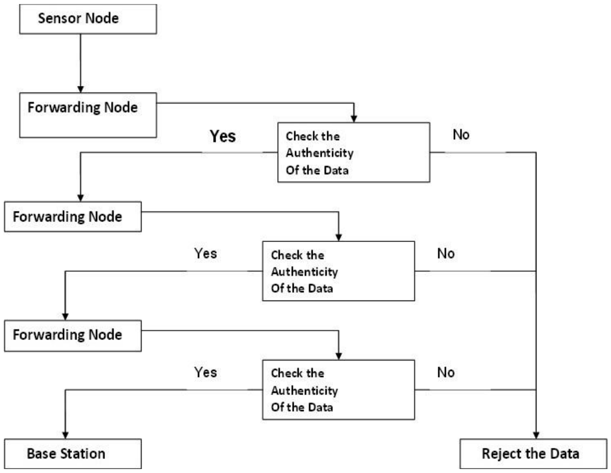

Sensor Node
Forwarding Node
Yes
Check the Authenticity Of the Data
No
Forwarding Node
Yes
Check the Authenticity Of the Data
No
Forwarding Node
Yes
Check the Authenticity Of the Data
No
Base Station
Reject the Data

Fig 2: Statistical En-Route Filtering

This strategy takes change of the noteworthy or strong utilization of remote sensor systems. Its finding and filtering power augmented through the utilization smallness and the sensor field measurement it can effectively identify false in arrangements even as the aggressor has obtained the assurance keys from a measure of placation center points. Given that key has a place with an undersized numeral of the key pool division. It can wipe out 80 to 90% false data through a participation hub inside 10 facilitating jumps. It described an essential stride in transit for building adaptable framework sensor arranges that can continue dealt center points. To stop any single exchange off center point beginning from the breaking the complete framework that method suspiciously limits the measure of security information designated to a specific hub and it relies on upon the public decisions of a few sensors for fake report location. While an occasion happens in the field that numerous close-by sensors all things considered produce a reasonable report that passes on some message acceptance codes. A report by a lacking measure of MACs won't be appropriated. Since a detecting portrayal is sent towards the sink more than complex bounces and it's each advancing hub confirmed of the exactness of the MACs acknowledged in the report with impacted likelihood. At the point when a stirred up MAC is distinguished then the report is fizzled. In the possibility of finding incorrectly MACs expands through the quantity of jumps the data wanders. Dependent upon the way span and there is a non-zero hazard in order to a couple reports with wrong MACs could escape on the way separating and be coursed to the sink. In any compartment the sink will extra verify the precision of Every MAC acknowledged in each report and dispose of false ones. Shared separating of false reports obliges to center points confer positive measure of wellbeing information. The extra security information each sending center point claims and the more beneficial the in transit channel could be except for activity is that if by one means or another more number of hubs is bargained, then the aggressor can accomplish more mystery from an exchanged off center.

C. Secure Ticket-Based En-Route Filtering

In STEF arranged through Krauss et al. organized through Krauss et al. uses a ticket thought some spot tickets are issued through the sink with groups are simply sent yet they hold a bonafide ticket. In any case a group does not encase a decent ticket and it is quickly filtered out. This system addresses false information infusion or DOS (Denial of Service) assault in sensor systems. This is a deficient ticket thought which is proper in sources watched remote sensor systems. A message toward the sink is basically appropriate yet it's including an authentic ticket. Each in transit center point which propels a message is proficient to approve the authenticity of this ticket or falls of the message in any case, the ticket is unsuitable. Thus, the fake messages may be isolated missing immediately. Furthermore, the ticket model enables this segment of report creation with sink affirmation and this in transit separating with no the necessity for symmetric key appointment around sensor center points. Despite the likelihood that restriction bargains a couple of center points is not skilled to imbue as bundles of messages as got a kick out of the chance to accomplish a fruitful Denial of administration assault since it isn't control the mandatory tickets. If a zone is under suspicion to be exchanged off, it may be viably banished through generally not sending request messages Containing quality tickets there, Moreover, are slanted toward the speedy enveloping region of the dealt center points and don't affect the entire system. Contemplating presentation, this strategy is proficient to essentially to widely lessen the vitality usage all through speedy filtering of false reports Also, the memory space necessities in the sensor center points is extremely discourteous, thus, it's applicable in hoisted thickness framework systems, notwithstanding leaves space for additional well-being instrument, which can add to the considered safeguard inside and out for the sensor system. In STEF (Secure Ticket-Based En-course Filtering) is parallel inside nature to SEF and DEF. The parcels incorporate a MAC and gathering heads segment keys by their quick source sensor centers in their area and around the sink. The negative division of STEF is its limited correspondence in the downstream for the ticket traversal toward the gathering head. .

D. Dynamic En-Route Filtering (DEF) Scheme

A legitimate packet is approved by multiple nodes using their own authentication keys in the Dynamic En-route Filtering (DEF) scheme [14]. Before deployment each node is preloaded with a seed authentication key and secret keys that are randomly chosen from a global key pool. The cluster head broadcasts authentication keys to en-route nodes encrypted with secret keys before sending the packet, that will be used for approval. If they can decrypt them successfully then reroute nodes store the keys. Each en-route node validates the integrity of the packet and drops the false ones. Consequently cluster heads send authentication keys to validate the packet. To spread the authentication keys, DEF method involves the usage of authentication keys and secret keys.

E. VEBEK: Virtual Energy-Based Encryption and Keying

For cyber physical network (CPN) VEBEK [15] is a secure network protocol. It uses one-time dynamic key generated by the source node for one packet, so it reduces the overhead of refreshing keys. Here to provide confidentiality of the data RC4 encryption mechanism is used. The key is generated from Virtual Energy based keying module for encryption. The receiving node must keep track of the energy of the sending node to decode and authenticate a message. It verifies its watch list to confirm that the packet came from a node it is watching when an en-route node receives the packet. The packet is forwarded without modification if verification fails. Two operational modes VEBEK-I and VEBEK-II are provided by VEBEK. All nodes watch their neighbors and when a packet is received from a neighboring node, its authenticity and integrity are verified in VEBEK-1 mode. It Can catch the malicious node in one hop itself and hence transmission overhead is minimized. But processing overhead is increased due to the decode/encode that occurs at each hop. Node in the network is organized to watch some of the nodes and it cannot find malicious packets in one hop, in VEBEK-II mode. More energy will be spent for node synchronization and this leads to overhead for the node.

F. Attacks on Cyber Physical Network

Here we present simple but previously neglected attacks on source routing protocols, such as DSR. In these systems, the source node specifies the entire route to a destination within the packet header, so intermediaries do not make independent forwarding decisions, relying rather on a route specified by the source. To forward a message, the intermediate node finds itself in the route (specified in the packet header) and transmits the message to the next hop. The burden is on the source to ensure that the route is valid at the time of sending, and that every node in the route is a physical neighbour of the previous route hop. This approach has the advantage of requiring very little forwarding logic at intermediate nodes, and allows for entire routes to be sender-authenticated using digital signatures, as in Ariadne.

We evaluated both the carousel and stretch attacks (Fig 3) in a randomly-generated 30-node topology and a single randomly-selected malicious DSR agent, using the Omnet++ network simulator. Energy usage is measured for the minimum number of packets required to deliver a single message, so sending more messages increases the strength of the attack linearly until bandwidth saturation.1 we independently computed resource utilization of honest and malicious nodes and found that malicious nodes did not use a disproportionate amount of energy in carrying out the attack. In other words, malicious nodes are not driving down the cumulative energy of the network purely by their own use of energy. Nevertheless, malicious node energy consumption data is omitted for clarity.The attacks are carried out by a randomly-selected adversary using the least intelligent attack strategy to obtain average expected damage estimates. More intelligent adversaries using more information about the network would be able to increase the strength of their attack by selecting destinations designed to maximize energy usage.

Per-node energy usage under both attacks is shown in Figure 4. As expected, the carousel attack causes excessive energy usage for a few nodes, since only nodes along a shorter path are affected. In contrast, the stretch attack shows more uniform energy consumption for all nodes in the network, since it lengthens the route, causing more nodes to process the packet. While both attacks significantly network-wide energy usage, individual nodes are also noticeably affected, with some losing almost 10% of their total energy reserve per message.Figure 3(a) diagrams the energy usage when node 0 sends a single packet to node 19 in an example network topology with only honest nodes. Black arrows denote the path of the packet.

Carousel attack: In this attack, an adversary sends a packet with a route composed as a series of loops, such that the same node appears in the route many times. This strategy can be used to increase the route length beyond the number of nodes in the network, only limited by the number of allowed entries in the source route. An example of this type of route is in Figure 1(a). In Figure 3(a), malicious node 0 carries out a carousel attack, sending a single message to node 19 (which does not have to be malicious). Note the drastic increase in energy usage along the original path.3 Assuming the adversary limits the transmission rate to avoid saturating the network, the theoretical limit of this attack is an energy usage increase factor of O (λ), where λ is the maximum route length.

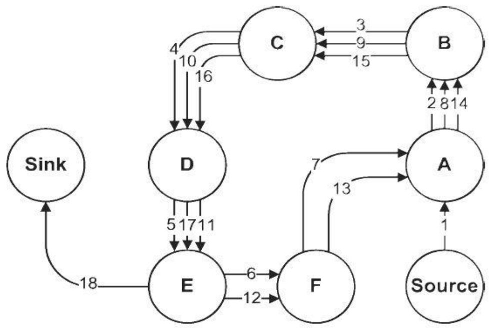


Fig. 3 (a) Carousal Attack

Overall energy consumption increases by up to a factor of 3.96 per message. On average, a randomly-located carousel attacker in our example topology can increase network energy consumption by a factor of 1.48 ± 0.99. The reason for this large standard deviation is that the attack does not always increase energy usage — the length of the adversarial path is a multiple of the honest path, which is in turn, affected by the position of the adversary in relation to the destination, so the adversary's position is important to the success of this attack.

Stretch attack: Another attack in the same vein is the stretch attack, where a malicious node constructs artificially long source routes, causing packets to traverse a larger than optimal number of nodes. An honest source would select the route Source → F → E → Sink, affecting four nodes including itself, but the malicious node selects a longer route, affecting all nodes in the network. These routes cause nodes that do not lie along the honest route to consume energy by forwarding packets they would not receive in honest scenarios.

An example of this type of route is in Figure 1(b). The outcome becomes clearer when we examine Figure 3(c) and compare to the carousel attack. While the latter uses energy at the nodes that were already in the honest path, the former extends the consumed energy "equivalence lines" to a wider section of the network. Energy usage is less localized around the original path, but more total energy is consumed.

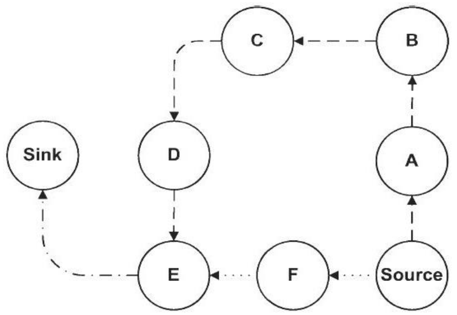


Fig. 3 (b) Stretch Attack

The theoretical limit of the stretch attack is a packet that traverses every network node, causing an energy usage increase of factor O(min(N, λ)), where N is the number of nodes in the network and λ is the maximum path length allowed.This attack is potentially less damaging per packet than the carousel attack, as the number of hops per packet is bounded by the number of network nodes. However, adversaries can combine carousel and stretch attacks to keep the packet in the network longer: the resulting "stretched cycle" could be traversed repeatedly in a loop. Therefore, even if stretch attack protection is not used, route loops should still be detected and removed to prevent the combined attack. In our example topology, we see an increase in energy usage by as much as a factor of 10.5 per message over the honest scenario, with an average increase in energy consumption of 2.67±2.49. As with the carousel attack, the reason for the large standard deviation is that the position of the adversarial node affects the strength of the attack. Not all routes can be significantly lengthened, depending on the location of the adversary. Unlike the carousel attack, where the relative positions of the source and sink are important, the stretch attack can achieve the same effectiveness independent of the attacker's network position relative to the destination, so the worst-case effect is far more likely to occur.The true significance of the attack becomes evident in Figure 4(a), which shows network-wide energy consumption in the presence of a single randomly-selected Vampire in terms of the "maliciousness" of the adversary, or the induced stretch of the optimal route in number of hops. (Increasing maliciousness beyond 9 has no effect due to the diameter of our test topology.) Network links become saturated at 10,000 messages per second (even without the stretch attack), but the adversary can achieve the same effects by sending an order of magnitude fewer messages at a stretch attack maliciousness level of 8 or greater. This reduces cumulative network energy by 3%, or almost the entire lifetime of a single node. Therefore, the stretch attack increases the effectiveness of an adversary by an order of magnitude, reducing its energy expenditure to compose and transmit messages. With 100 messages, the result is less severe, but still pronounced: the network loses 1% of its total energy, or 9% of the lifetime of a single node. The effect becomes less visible when we look at 10 messages or fewer in Figure 4(b), but is still noticeable.

Since DSR uses hop count as a cost metric, constructing longer source routes could in fact decrease the amount of per hop energy spent on sending packets if energy minimization protocols were used since shorter physical distances decrease required sending power, and thus battery drain. We construct long routes greedily, assuming global topology knowledge, but attacks can be further optimized to consume more energy by considering relative node distances — given enough information, our adversary could construct not just longer but maximum-energy paths. Forwarding nodes using minimum energy routing could replace long distance transmissions with a number of shorter-distance hops, but the attack still works since the malicious path is longer, independent of in-network optimizations applied to it.

These attacks would be less effective in hierarchical networks, there nodes send messages to aggregators, who in turn send it to other aggregators, which route it to a monitoring point. The described attacks are only valid within the network "neighbourhood" of the adversarial node. If an adversary corrupts nodes intelligently or controls a small but non-trivial percentage of nodes, it can execute these attacks within individual network neighbourhood: a single adversary per neighbourhood would disable the entire network

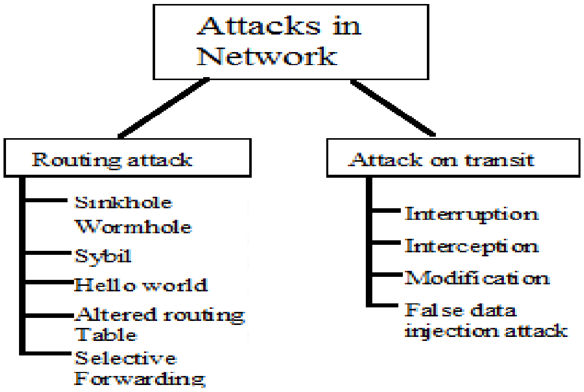


Fig 4. Attacks in CPNS

IV. Methodology

We now depict the crucial thought about our arrangement. PCREF uses polynomials instead of MACs to verify reports, and can mitigate the node impersonating attack against legitimate nodes. By sorting out a plan of identifying center points into a gathering, where center points are responsible for the same checked portions, PCREF apportions the looking at acceptance polynomial in addition; check polynomials to each sensor center point. These polynomials set away in center points are bundled with center ID and gathered by the primitive polynomials doled out from a primitive polynomial pool. Unmistakable primitive polynomials will be used as a piece of different gatherings through the group based primitive polynomial undertaking. In the gathering based primitive polynomial errand, center points in different gatherings are apportioned particular primitive polynomials from the overall primitive polynomial pool and make assorted confirmation polynomial likewise, check polynomial.

In perspective of this errand, paying little respect to the likelihood that foes exchange off center points in one gathering, they won't impact the security of center points in various gatherings. The affirmation polynomial set away in each center is used to bolster the report of adjacent portion estimation while the check polynomial is used to favor the got reports. Every identifying center stores the affirmation polynomial of the close-by cluster and stores the check polynomial of various gatherings with a pre-portrayed probability. Each sending center stores the check polynomial of each gathering with the same probability. Note that, probability is familiar with measure the attractive probability of sharing affirmation information between two centers in CPNS and can impact the authenticity of PCREF, the distinct examination can be found. Our arrangement moreover uses - acceptance framework like [13]–[17], i.e., a honest to goodness report ought to be confirmed by center points from

the same group. Our arrangement involves the going with two key sections:
A) Confirmation information organization is used to dole out the key, check polynomial, check polynomial, and close-by ID of recognizing centers.
B) Data security organization is used to recognize and channel the false estimation reports. These two sections will be depicted in the accompanying two subsections.

V. DESIGN/ IMPLEMENTATION

STEP 1: Designing of Network

Here, Simulator named OMNeT++ is used for designing network. OmneT++ is an object-oriented modular discrete event network simulation framework. It has a generic architecture, so it can be (and has been) used in various problem domains:

1.modeling of wired and wireless communication networks
2.protocol modeling
3.modeling of queueing networks
4. modeling of multiprocessors and other distributed hardware systems
5. validating of hardware architectures
6. evaluating performance aspects of complex software systems
7. in general, modeling and simulation of any system where the discrete event approach is suitable, and can be conveniently mapped into entities communicating by exchanging messages.

OMNeT++ itself is not a simulator of anything concrete, but rather provides infrastructure and tools for writing simulations. One of the fundamental ingredients of this infrastructure is a component architecture for simulation models. Models are assembled from reusable components termed modules.

Modules can be connected with each other via gates (other systems would call them ports), and combined to form compound modules. The depth of module nesting is not limited. Modules communicate through message passing, where messages may carry arbitrary data structures. Modules can pass messages along predefined paths via gates and connections, or directly to their destination; the latter is useful for wireless simulations, for example. Modules may have parameters that can be used to customize module behavior and/or to parameterize the model's topology. Modules at the lowest level of the module hierarchy are called simple modules, and they encapsulate model behavior. Simple modules are programmed in C++, and make use of the simulation library. An OMNeT++ model consists of modules that communicate with message passing. The active modules are termed simple modules; they are written in C++, using the simulation class library. Simple modules can be grouped into compound modules and so forth; the number of hierarchy levels is unlimited. The whole model, called network in OMNeT++, is itself a compound module. Messages can be sent either via connections that span modules or directly to other modules.

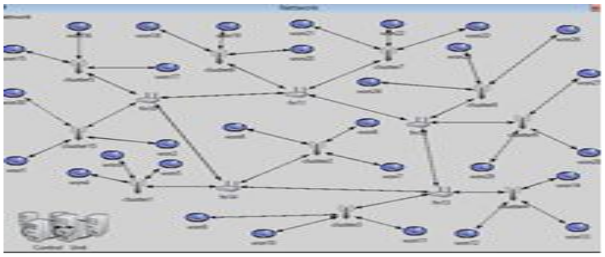


Fig.5 Design of Network

STEP 2: Naming And Assigning Id’s

We describes the structure of a simulation model in the NED language. NED stands for Network Description. NED lets the user declare simple modules, and connect and assemble them into compound modules. The user can label some compound modules as networks; that is, self-contained simulation models. Channels are another component type, whose instances can also be used in compound modules.

The NED language has several features for large projects

Hierarchical: The traditional way to deal with complexity is by introducing hierarchies. In OMNeT++, any module which would be too complex as a single entity can be broken down into smaller modules, and used as a compound module.

Component-Based: Simple modules and compound modules are inherently reusable, which not only reduces code copying, but more importantly, allows component libraries (like the INET Framework, MiXiM, Castalia, etc.) to exist.

Interfaces: Module and channel interfaces can be used as a placeholder where normally a module or channel type would be used, and the concrete module or channel type is determined at network setup time by a parameter. Concrete module types have to “implement” the interface they can substitute. For example, given a compound module type named MobileHost contains a mobility submodule of the type IMobility (where IMobility is a module interface), the actual type of mobility may be chosen from the module types that implemented IMobility (RandomWalkMobility, TurtleMobility, etc.)

Inheritance: Modules and channels can be subclassed. Derived modules and channels may add new parameters, gates, and (in the case of compound modules) new submodules and connections. They may set existing parameters to a specific value, and also set the gate size of a gate vector. This makes it possible, for example, to take a GenericTCPClientApp module and derive an FTPClientAppfrom it by setting certain parameters to a fixed value; or to derive a WebClientHost compound module from a BaseHost compound module by adding a WebClientApp submodule and connecting it to the inherited TCP submodule.

Packages: The NED language features a Java-like package structure, to reduce the risk of name clashes between different models. NEDPATH (similar to Java's CLASSPATH) has also been introduced to make it easier to specify dependencies among simulation models.

Inner types: Channel types and module types used locally by a compound module can be defined within the compound module, in order to reduce namespace pollution.

Metadata annotations. It is possible to annotate module or channel types, parameters, gates and submodules by adding properties. Metadata are not used by the simulation kernel directly, but they can carry extra information for various tools, the runtime environment, or even for other modules in the model. For example, a module's graphical representation (icon, etc) or the prompt string and measurement unit (milliwatt, etc) of a parameter are already specified as metadata annotations.

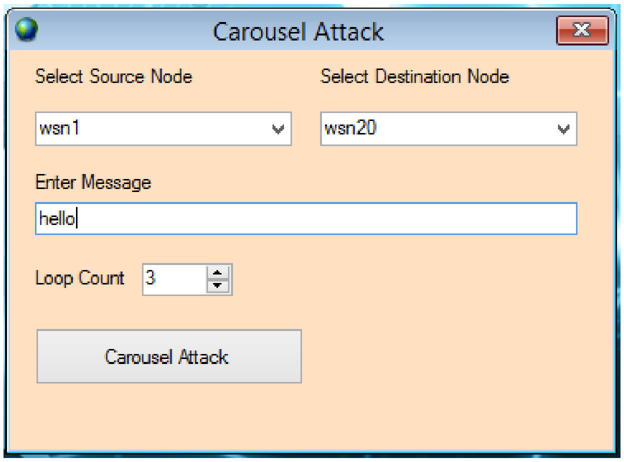
STEP 3: Implementation of Attacks

We have implemented three attacks in our system named as:

1. Carousel Attack
2. Stretch Attack
3. False Data Injection

Carousel Attack:

In this attack, an adversary sends a packet with a route composed as a series of loops, such that the same node appears in the route many times. This strategy can be used to increase the route length beyond the number of nodes in the network, only limited by the number of allowed entries in the source route



Carousel Attack

Select Source Node: wsn1

Select Destination Node: wsn20

Enter Message: hello

Loop Count: 3

Fig.6: Carousel Attack

Malicious node 0 carries out a carousel attack, sending a single message to node 19 (which does not have to be malicious). Note the drastic increase in energy usage along the original path. Assuming the adversary limits the transmission rate to avoid saturating the network, the

VI. Result & Conclusion

We have used Omnet++ Network Simulator to simulate our system. The Implementation is divided in mainly three phases, i) Local Id Assignment, ii) Attacks and iii) Prevention. In this project, we proposed a system, which can channel filter false information on the way successfully and accomplish high flexibility to the quantity of traded off hubs without depending on static courses and hub confinement. This system embraces polynomials for underwriting estimation reports to enhance strength to hub imitating assaults. Every hub stores two sorts of polynomials:

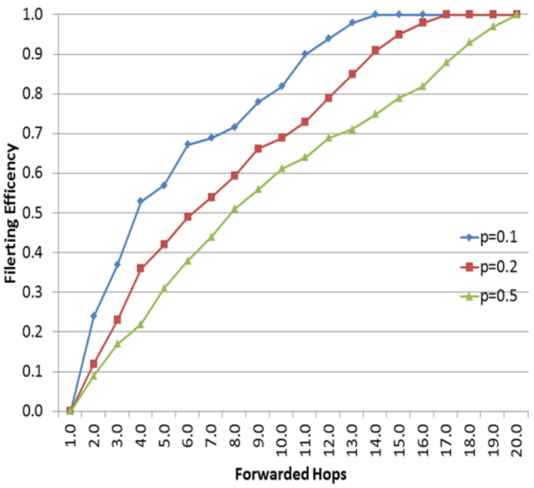


Fig.7: Proposed System

The filtering effectiveness of PCREF can be spoken to by, which is characterized as the likelihood of false estimation answer to be separated inside various jumps. The more noteworthy the likelihood, the better the sifting effectiveness progresses toward becoming. Numerical outcomes in Fig. 3 demonstrate the sifting effectiveness versus the sent bounces when P=0.1, 0.2, 0.5. As should be obvious, PCREF can channel a large portion of the false estimation reports amid the steering way, and therefore it can distinguish and channel false estimation reports successfully. The higher the estimation of , the littler the quantity of sent bounces is required to channel the false estimation reports. This is on the grounds that the likelihood of the check polynomial put away at the middle of the road hub increments as P increments. Be that as it may, each middle of the road hub stores (Ns/n).P check polynomials, and a littler P can decrease the capacity overhead of the halfway hub.

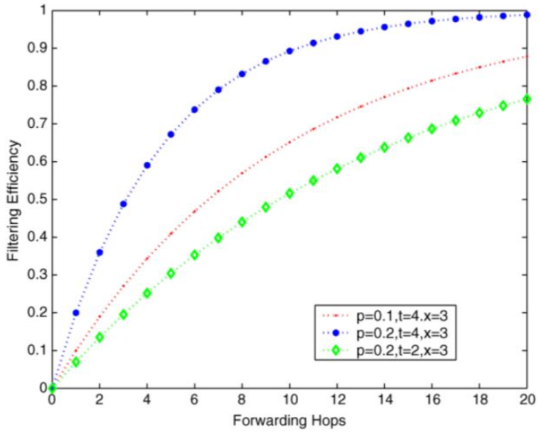


Fig.8-Existing System

VII. FUTURE SCOPE

To mitigate the false data injected by the adversary in sensor networks, a number of en-route filtering schemes have been developed . For example SEF and IHA are the first two proposed schemes to conduct en-route filtering of false reports. Both SEF and IHA have the -threshold limitation. All these limit their usage in CPNS.

So we can do further study to overcome this limitation to improve the security level of CPNS.

In the future, we are planning to test network of wireless sensors and study the behavior of the algorithm in a real-life setting. We will also make several extensions to the protocol so that it can benefit from extra information on the sensors' mobility patterns and mobility-pattern variability. This encompasses maintaining knowledge about sensors' speed and direction, possibly using additional equipment such as accelerometers and deploying it in heterogeneous networks using a mix of both mobile and static anchors.

REFERENCES

[1] Xinyu Yang, Jie Lin, Wei Yu, Paul-Marie Moulema, Xinwen Fu, and Wei ZhaoA, "Novel En-Route Filtering Scheme Against False Data Injection Attacks in Cyber-Physical Networked Systems", in IEEE TRANSACTIONS ON COMPUTERS, VOL. 64, NO. 1, JANUARY 2015

[2] F. Wu, Y. Kao, and Y. Tseng, "From wireless sensor networks towards cyber physical systems" , in Pervasive Mobile Comput, vol. 7,no. 4, pp. 397–413, Aug. 2011.

[3] Q. Yang, J. Yang, W. Yu, N. Zhang, and W. Zhao, "On a hierarchical false data injection attack on power system state estimation", in Proc. IEEE Global Telecommun. Conf. (GLOBECOM'11), 2011.

[4] Qingyu Yang, Member, IEEE, Jie Yang, Wei Yu, Dou An, Nan Zhang, and Wei Zhao, Fellow, IEEE "False data injection attacks against state estimation in electric power grids", in IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 3, MARCH 2014

[5] Y. Mo and B. Sinopoli, "False data injection attacks in control systems", in Proc. Preprints 1st Workshop Secure Control Syst., CPS Week, 2010.

[6] Z. Yu and Y. Guan, "A dynamic en-route filtering scheme for data reporting in wireless sensor networks", in IEEE/ACM Trans.Networking (ToN), vol. 18, pp. 150–163, 2010.

[7] Y.-S. Chen and C.-L. Lei, "Filtering false messages en-route in wireless multi-hop networks", in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), 2010.

[8] Ms. Kanchan Babanrao Korke, Prof V. K. Barbudhe, "A Case study of En-Route Filtering Scheme Against False Data Injection Attacks in Cyber Physical Networked Systems" in International Journal on Recent and Innovation Trends in Computing and Communication,Volume-4,Issue-1,January 2016.