I.J. Information Technology and Computer Science, **2012, 6, 45-56**
Published Online June 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2012.06.07

Dynamic High Level Cross Layer Security Mechanisms for Wireless Sensor Networks

Sanjeev Puri
Research Scholar, Singhania University, Jhunjhunu**,**
Email: purispuri_2005@rediffmail.com

S.P Tripathi
Professor, Institute of Engineering &Technology, Lucknow
Email: tripathe_sp@yahoo.co.in

***Abstract*—** In dynamic insensitive application specific network, WSN consists of hundred and thousands of sensor nodes densely deployed in the sensor field. Majority of sensor nodes are static having power limitations, low network throughput, message transfer delays and computation power limits that are major obstacles. The limited communication range of WSN nodes, link asymmetry, and the characteristics of the physical environment lead to a major sources of QoS degradation in WSNs. The potential applications of the WSNs typically range from those in defense, military, environmental monitoring, health monitoring and civilian surveillance applications etc. All of these applications being omnipresent in nature necessitate appropriate heavy secure mechanisms to ensure the data security and privacy. On the other hand, the WSN nodes, being extremely resource constrained, it is really challenging to devise the WSNs security protocols. So need to propose dynamic high level cross layer security mechanism for detecting various attacks and the countermeasures taken to avoid the same without comprising any network resources.

***Index Terms*—** physical environments, QoS metrics, WSN nodes, dynamic cross layer security, secure mechanisms, WSN security protocols

I. Introduction

Security can be considered as a non-functional requirement that maintains the overall system usable and reliable, protecting the information and information systems. In fact, in wireless sensor networks, security is of paramount importance. Attackers can eavesdrop on our radio transmissions, inject bits in the channel, replay previously heard packets and many more.

In this paper, we discuss the classes of various attacks against sensor networks, threat models and security goals for secure routing in wireless sensor networks. Then the countermeasures and designs considerations us for secure routing protocols in sensor networks. It is unlikely a sensor network routing protocol can be made secure by incorporating security mechanisms after design has completed. Secure routing is vital to the acceptance and use of sensor networks for many applications, but we have demonstrated that currently proposed routing protocols for these networks are insecure. So leave it as open research problems to cross-layer design of a sensor network routing protocol that satisfies the future WSN security goals.

Research problems happen in the expansion of organized practice for cross-layer design of WSN routing protocols for the pass on applications. Cross-layer design enables different layers of the communication stack to share entire information and coordinate their actions. In this context, we depict and analyze the impact as well as consequent risks of a cross-layer approach towards get dynamic research directions [1]. A cross-layer solution, in fact, generally decreases the level of modularity, which may lose the decoupling between design and development process, making it more difficult to further design improvements and innovations.

There are several open research problems toward the development of methodical techniques for cross-layer design of wireless sensor network protocols:

1) **Improved energy consumption and maximum network lifetime:** Existing studies on cross-layer optimization are mostly focused on jointly optimizing functionalities at different layers, usually with the overall objective of maximizing the network throughput. Conversely, in WSNs the ultimate objective is usually to minimize the energy consumption and/or to maximize the network lifetime. Hence, further study is needed to develop models and methodologies suitable to solve energy-oriented problems. There is a need to develop sound models to include in the above framework an accurate description of the end-to-end delay as results from the interaction of the different layers. This is particularly important for the design of sensor network protocols for monitoring applications that require real-time delivery of event data, such as those encountered in wireless sensor and actor networks.

2) **Connectivity with realistic physical layer:** However, recent experimental studies have demonstrated that the effects of the impairments of the wireless channel on higher-layer protocols are not negligible, as the availability of links further fluctuates because of channel fading phenomena that affect the wireless transmission medium. Furthermore, mobility of nodes is not considered. In fact, due to node mobility and node join and leave events, the network may be subject to frequent topological reconfigurations. Thus, links are continuously established and broken. For the above reasons, new analytical models are required to determine connectivity conditions that incorporate mobility and fading channels.
3) **Redundant security provisioning:** Unfortunately, there may be several protocol layers within the network protocol stack which are capable of providing security services to the same attack. Consequently, when the original data go through the protocol stack starting from the highest layer, they will be processed layer-by-layer. To this end, some part of the data packets may go through the security-prerequisite operations of different layers and result in *redundant security* provisioning.
4) **Security schemes problems:** For instance, link layer security scheme typically addresses confidentiality (data privacy) provisioning, authentication (source and data integrity) and data freshness, but no security issues in the physical layer. However, an insecure physical layer may practically make the entire network remain insecure. So, it is easy to figure out that cross-layer solutions can accomplish better performance. Furthermore, an additional security capability can be achieved via self-adaptive security services, because they are flexible in dealing with the dynamic network topology as well as different types of attacks.
5) **Cross-layer discrete-event network simulators:** Current discrete-event network simulators such as OPNET, NS-2, J-Sim and GloMoSim may be unsuitable to implement a cross-layer solution, since their inner structure is based on a layered architecture, and each has implemented functionality run by the simulator engine is tightly tied to this architecture. Hence, implementing a cross-layer solution in one of these simulators may turn into a non-trivial task. For this reason, there is a need to develop new software simulators that are based on a new developing paradigm so as to ease the development and test of cross-layer algorithmic and protocol solutions.

To simplify the problems, following sub-problems are identified in following manner i.e.

To define WSN protocol architecture that can explicitly accommodate cross layer design and optimization issues. The lack of standard architecture prohibits software reusability resulting in waste of time, effort, and money. Also the existing architectures do not support the cross layer design explicitly and therefore, the benefits that one can achieve from cross layer information exchange cannot be achieved. So the task is to define a multi-hop multi scale WSN architecture which supports cross layer approach and provides plug-and-play features at the same time. The proposed solution utilizes a *feedback-based congestion control* to guarantee packet delivery speed across the network. With such a support, applications can estimate an end-to-end delay before making admission decisions and dynamically adjust the workload they generate to meet their real-time requirements. [1][2]

An enhance scheme to incorporate real-time guarantees and differentiated QoS supports into this aggregation framework. Our solution is expected to improve the efficiency in bandwidth utilization, a resource that is most precious in sensor networks and the energy-conservation by reducing packet collisions and control overhead. Treating the entire communication protocol stack in a holistic manner can help in finding new means to alleviate the harmful performance restraining consequences of common wireless network problems, such as burst errors due to channel distortions, wireless interference problems, multipath propagation or fading effects. Acknowledging that state-based solutions are inefficient to cope with highly dynamical sensor networks, intend a solution that is altogether state-free for robust data delivery. In this solution, we aim at providing not only a reliable communication scheme, but also a fast response and recovery from the failures with a much less control overhead.

There is must to focus on a swift & self-stabilizing approach to deal with instability caused by fast flow dynamics inside networks such as nodes' failure and mobility. A efficient approach to reduce the inconsistency between outdated routing information a node keeps and the volatile network situations with minimal overhead A reliable scheme which prevents the performance degradations in packet delivery, end-to-end delay and control overhead, while allowing nodes going to a dormant state in order to conserve energy effectively.

Localization techniques in sensor networks can be divided into to two major categories: range-based localization and range-free localization. Range-based localizations are widely investigated in recent years. Such technique yells better precision under control environment or by using sophisticated devices. Much less research has be done on range-free localization, which are regard as an cost-effective and sufficient solution for sensor networks without costly hardware requirements. Fairly need to design a scalable localization algorithm with enhanced performance over pervious solutions.

A blind collision occurs when two nodes, which are not visible to each other due to limited transmission range, presence of asymmetric links, presence of

obstacles, etc., communicate with a commonly visible node during a given time interval. This leads to the degradation of the following three performance metrics. First is *Throughput*, which denotes the amount of traffic successfully received by a destination node and that decreases due to additional blind collisions. Second is *Energy-efficiency* that decreases since each collision causes a new retransmission. Third is *Transfer delay*, which represents the time duration from the generation of a message until its correct reception by the destination node, and that becomes larger due to the multiple retransmissions of a collided message.

Beside the problems of battery power, QoS routing, MAC scheduling, and efficient utilization of network resources, multi-hop wireless networks are more vulnerable to different security risks due to inherent attack prone features such as shared MAC, multi-hop decentralized architecture, wireless medium etc.[3] The attackers can exploit these features to bring serious disorders and routing disruption. Furthermore, multi-hop wireless networks are exposed to multi-layer threats. Hence cross layer security mechanisms are indeed necessary for this adaptive scalable WSN design.

II. Wireless Sensor Networks Security Analysis

Securing the Wireless Sensor Network needs to make the network support all security properties: confidentiality, integrity, authenticity and availability [1]. Attackers may deploy a few malicious nodes with similar hardware capabilities as the legitimate nodes that might collude to attack the system cooperatively. Sensor nodes may not be tamper resistant and if an adversary compromises a node, we can extract all key material, data, and code stored on that node. While tamper resistance might be a viable defense for physical node compromise for some networks, we do not see it as a general purpose solution. Extremely effective tamper resistance tends to add significant per-unit cost, and sensor nodes are intended to be very inexpensive.

A. Categories of Security threats

The network must be adequately protected against malicious threats that can affect its functionality. Due to the role of sensor networks as a sensory system", any disturbance in a sensor network may have consequences in the real world. However, achieving this goal is not an easy task, because sensor networks are especially vulnerable against external and internal attacks due to their peculiar characteristics. The devices of the network i.e. sensor nodes are highly constrained in terms of computational capabilities, memory, and communication bandwidth and battery power. Additionally, it is easy to physically access such nodes because they must be located near the physical source of the events, and they usually are not tamper-resistant due to cost constraints. Furthermore, any internal or external device can access to the information exchange because the communication channel is public. As a result, sensor networks have to face multiple threats that may easily hinder its functionality and nullify the benefits of using its services. [4] These threats to WSN can be categorized as follows:

- Common attacks
- Denial of service attack
- Node compromise
- Impersonation attack
- Protocol-Specific attacks

Common attacks - Due to sensor networks nature, there are some *specific attacks* targeting the communication channels. An adversary can easily retrieve valuable data from the transmitted packets that are sent (Eavesdropping). That adversary can also simply intercept and modify the packets' content meant for the base station or intermediate nodes (Message Modification), or re-transmit the contents of those packets at a later time (Message Replay). Finally, the attacker can send out false data into the network, may be masquerading as one of the nodes, with the objectives of corrupting the collected sensors' reading or disrupting the internal control data (Message Injection). Since sensor networks are wireless service-oriented infrastructures.

Denial of service attack - Denial of Service (DoS) is any event that diminishes or eliminates a network's capacity to perform its expected function. A DoS attack on a WSN may take several forms [5]. The most important are the following: Node collaboration, in which a set of sensor nodes act maliciously and prevent broadcast messages from reaching certain section(s) of the sensor network, jamming attack, in which an attacker jams the communication channel and avoid any member of the network in the affected area to send or receive any packet, and exhaustion of power, in which an attacker repeatedly requests packets from nodes to deplete their battery life.

Node compromise - A sensor node is considered as being compromised when an attacker, through various means, can either read or modify its internal memory. Attacks can be invasive or non-invasive. An invasive physical attack is defined as an attack where the attacker physically breaks into the hardware by modifying its hardware structure (e.g. using focused ion beam, or drilling a hole in the storage media). On the other hand, a non invasive attack is defined as an attack where the data is taken from the hardware device without any form of structural modification done to the device. Various complex attacks can be easily launched from compromised sensor nodes, since the subverted node is a full-fledged member of the network.

Impersonation Attacks - The most common attacks to compromised node are the *impersonation attacks*. In an impersonation attack, a malicious node impersonates a legitimate node, and uses its identity to mount active attacks such as Sybil attacks or node replication attacks. In a Sybil attack, a single sensor node takes on multiple

identities to deceive other sensor nodes. A sensor node that wishes to conduct the Sybil attack can adopt a new identity by creating a new identity or by stealing the identity of an existing sensor node. On the other hand, node or identity replication is the simple duplication of sensor nodes. As sensor nodes tend to be physically unprotected, it is feasible for an attacker to capture, replicate and insert duplicate nodes back into selected regions of the network. Node replication is different from a Sybil attack in that the multiple sensor nodes are duplicates and basically have the same identities.

Attacks against routing protocols - The Attacks against routing protocols in a WSN are corruption of the internal control information such as the routing tables (Spoofed Routing Information), selective forwarding of the packets that traverse a malicious node depending on some criteria (Selective Forwarding), creation of a "wormhole" that captures the information at one location and replays them in another location either unchanged (Wormhole attack) or tampered (Sinkhole attack), creation of false control packets during the deployment of the network (Hello Flood Attack), and creation of false acknowledge information (Acknowledgment Spoofing). Other protocols can be attacked as well, such as data aggregation e.g. by forging the data before, during after the aggregation process. [6]

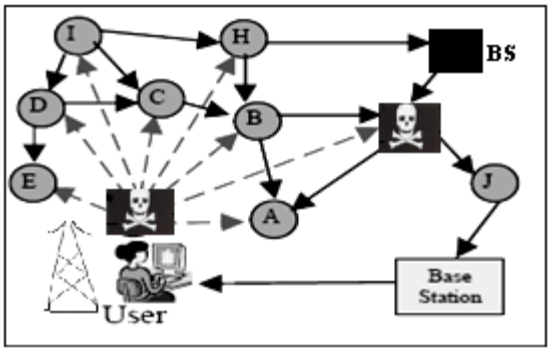

Fig 1: HELLO flood attacks Vs Wormholes attacks

Once sources begin to generate data events, an adversary attacking a data flow might have one of four goals: *Flow suppression* is an instance of denial of-service. The easiest way to suppress a flow is to spoof negative reinforcements. *Cloning* a flow enables eavesdropping. After an adversary receives an interest flooded from a legitimate base station, it can simply replay that interest with himself listed as a base station. An adversary can *path influence* taken by a data flow by spoofing positive and negative reinforcements and bogus data events.

B. Dynamic Security Mechanisms

Sensor networks are vulnerable to external and internal attacks. The effects of those attacks in the network are not trivial, since they can render the services of the network useless. There is the need of using security mechanisms either to prevent the attacks from influencing over the functionality of the network or to minimize the adverse effects of such attacks. By using the dynamic security mechanisms to enforce in sensor networks the following security properties [7]:

Authentication: The information received by the sensor nodes and the base station must come from a valid member of the network.

Authorization: Only authorized entities (sensor nodes and base station) can be involved in providing information to the network.

Confidentiality: A given message must not be understood by anyone other than the desired recipients.

Integrity: The data produced and consumed by the sensor network must not be maliciously altered.

Dynamic fault tolerance: Dynamic securities services in the presence of integrate faults such as failed arise in dynamic nodes.

Energetic ease of use: The users must be capable of access its services while they require it.

Dynamic secure link connectivity: The data produced by the sensor network must be recent.

Secure Topology & Self-organize: Every sensor node must be independent and flexible enough to self-organize and self-heal itself according to different situations.

In a sensor network context, the existing security mechanisms try to protect the hardware of the sensor nodes, the communication channel, and the protocols and services. By protecting the hardware of the sensor nodes, it is possible to detect and/or prevent attacks that try to compromise a sensor node. A secure communication channel cannot be affected by most of the common attacks (eavesdropping, modification, replay, and injection) that affect the exchange of messages between the sensor nodes. Finally, with the adequate support, the protocols and services used in the network can tolerate the existence of dynamic service disruptions and complex attacks.

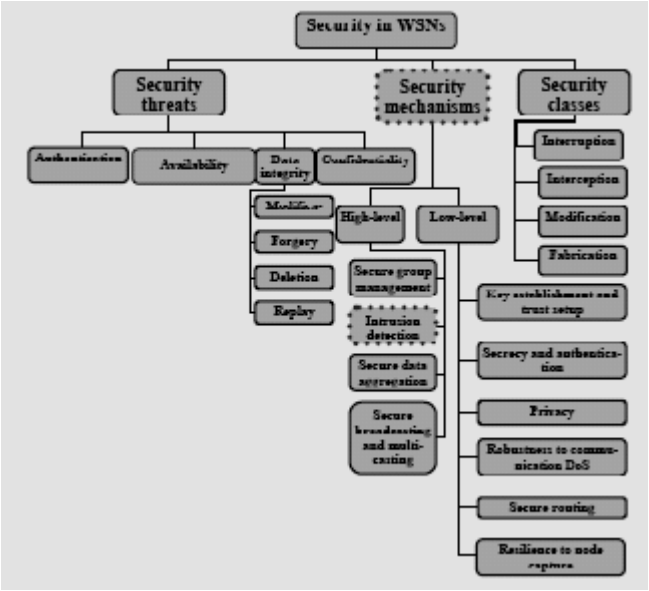

Fig 2: Security Mechanisms in WSN

1) Hardware protection- Once the attacker attains, threaten, and takes control over a sensor node, it can access to its internal information, and also use it for malicious purposes by launching complex or stealthy attacks. Therefore, there should be some kind of protection on the hardware layer to avoid such attacks, like a tamper proof module. Such modules allow the security credentials to be stored securely, preventing an attacker from retrieving the credentials when the sensor node is compromised. Once a tampering of the chip is detected, this module will destroy the keys and other information stored in the module. Unfortunately, although adding this module into a sensor node would help to defend against node compromise, the addition of the module will also significantly increase its overall cost. As a result, it is necessary to use other software-based mechanisms that are not dependent on any hardware configuration. Code attestation techniques can not directly defend against node compromise, but they can be able to detect whether a certain node has been compromised or not. Also, code obfuscation techniques increase the complexity of analyzing the memory of a node [8].

Code attestation- Software based attestation enables a third party to verify the code running on the system to detect any maliciously altered code. Usually code attestation is done through the use of special hardware mechanisms proposed by the Trusted Computing Group and Next Generation Secure Computing Based. Thus this kind of software attestation is designed to provide the detection of malicious code alteration and verify that the nodes are using the correct codes. A verification procedure is needed to effectively verify that the node's code is correct and not maliciously altered. A verifier needs to generate a random challenge to be sent to the node. The node will then use this challenge to generate a response using the verification procedure. The verifier will then compare the response against an expected value.

Any discrepancy would imply that the code in the node has been modified. Data used in the code attestation usually includes the clock speed, instruction set architecture, the memory architecture of the microcontroller and the size of the device's memory. The verification procedure is mostly based on the pseudorandom memory traversal concept: using a seed i.e. the random challenge provided by the verifier, the node must randomly access some positions of its own memory, summarizing them into one report that will be used as a response. If the node is malicious, the time used on calculating a valid response will be longer, and such delay can be detected by the verifier.

Code obfuscation- Code obfuscation, or diversification, is a mechanism that allows the protection of a valuable piece of information i.e. the security credentials contained inside the node. By obfuscating the code and data, the amount of time needed by the attacker to analyze the compromised nodes will increase, thus it will be more difficult to deduce the secrets from the extracted contents of program flash, the EEPROM or the SRAM. The obfuscation methods must not be equal for all the nodes. This is to prevent the attacker from using the same method to retrieve the secrets once he/she is successful in compromising one node. Those diversification techniques may include stack randomization, instruction set randomization, library randomization, and system call randomization. In a sensor node, it is possible to hide vital information, such as the secret keys, using a hash function to scramble the information in the data segment. By hiding the keys in a randomized manner, it would be difficult for the attacker to find the keys from the downloaded EEPROM.

2) Dynamic Selective Forwarding & Authenticated Broadcast - Even in protocols completely resistant to sinkholes, wormholes, and the Sybil attack, a compromised node has a significant probability of including itself on a data flow to launch a selective forwarding attack if it is strategically located near the source or a base station. Multipath routing can be used to counter these types of selective forwarding attacks. Messages routed over paths whose nodes are completely disjoint are completely protected against selective forwarding attacks involving at most compromised nodes and still offer some probabilistic protection whenever nodes are compromised. However, completely disjoint paths may be difficult to create. Braided paths may have nodes in common, but have no links in common. The use of multiple braided paths may provide probabilistic protection against selective forwarding and use only localized information. Allowing nodes to dynamically choose a packet's next hop probabilistically from a set of possible candidates can further reduce the chances of an adverse control of a data flow.

If we have base stations trustworthy, adversaries must not be able to spoof broadcast or *flooded messages* from any base station. This requires some level of asymmetry: since every node in the network can potentially be compromised, no node should be able to spoof messages from a base station, yet every node should be able to verify them. Authenticated broadcast is also useful for localized node interactions. Many protocols require nodes to broadcast HELLO messages to their neighbors. These messages should be authenticated and impossible to spoof. Proposals for authenticated broadcast intended for use in a more conventional setting either use digital signatures and/or have packet overhead that well exceed the length of typical sensor network packet. TESLA is a protocol for efficient, authenticated broadcast and flooding that uses only symmetric key cryptography and requires minimal packet overhead. SPIN and gossiping algorithms are techniques to reduce the messaging costs and collisions

which still achieve robust probabilistic dissemination of messages to every node in the network. [9]

III. Cross layer Counter Measures

In Cross-layer design, at physical layer, power can be automatically adjust with the interference potency, which ease energy consumption and strive to congest attacks. At MAC layer, we can reduce the number of retransmissions' packets, which in turn hold back fatigue attack and save energy as well. At the network layer, we can follow multi-path routing, which bypasses routing black-hole and ease the energy consumption due to congestion. So we can plot one vital approach to develop cross-layer based security mechanisms alone for diverse kinds of security counter measures [10] [11].

A. Link layer security

The majority of outsider attacks against sensor network routing protocols can be prevented by simple link layer encryption and authentication using a globally shared key. Major classes of attacks not countered by link layer encryption and authentication mechanisms are wormhole attacks and HELLO flood attacks because, although an adversary is prevented from joining the network, nothing prevents her from using a wormhole to tunnel packets sent by legitimate nodes in one part of the network to legitimate nodes in another part to convince them.

Link layer security mechanisms using a globally shared key are completely ineffective in presence of insider attacks or compromised nodes. Insiders can attack the network by spoofing or injecting bogus routing information, creating sinkholes, selectively forwarding packets, using the Sybil attack, and broadcasting HELLO floods. More sophisticated defense mechanisms are needed to provide reasonable protection against wormholes and insider attacks.

B. Prevent from Network layer attacks

An insider cannot be prevented from participating in the network, but he should only be able to do so using the identities of the nodes he has compromised. Using a globally shared key allows an insider to masquerade as any node. Identities must be verified. In the traditional setting, this might be done using public key cryptography, but generating and verifying digital signatures is beyond the capabilities of sensor nodes. One solution is to have every node share a unique symmetric key with a trusted base station. Two nodes can then use a Needham-Schroeder like protocol to verify each other's identity and establish a shared key. A pair of neighboring nodes can use the resulting key to implement an authenticated, encrypted link between them.

In order to prevent an insider from wandering around a stationary network and establishing shared keys with every node in the network, the base station can reasonably limit the number of neighbors a node is allowed to have and send an error message when a node exceeds it. Thus, when a node is compromised, it is restricted to communicating only with its verified neighbors. This is not to say that nodes are forbidden from sending messages to base stations or aggregation points multiple hops away, but they are restricted from using any node except their verified neighbors to do so. In addition, an adversary can still use a wormhole to create an artificial link between two nodes to convince them they are neighbors, but the adversary will not be able to eavesdrop on or modify any future communications between them.

The straightforward defense against HELLO flood attacks is to verify the bi-directionality of a link before taking meaningful action based on a message received over that link. The identity verification protocol is sufficient to prevent HELLO flood attacks. Not only does it verify the bidirectional link between two nodes, but even if a well-funded adversary had a highly sensitive receiver or had wormholes to a multiple locations in the network, a trusted base station that limits the number of verified neighbors for each node will still prevent HELLO flood attacks on large segments of the network when a small number of nodes have been compromised.

Wormhole and sinkhole attacks are very fiddly to defend against, especially when the two are used in combination. Wormholes are hard to detect because they use a private, out-of-band channel invisible to the underlying sensor network. Sinkholes are difficult to defend against in protocols that use advertised information such as remaining energy or an estimate of end-to-end reliability to construct a routing topology because this information is hard to verify. Routes that minimize the hop-count to a base station are easier to verify, however hop-count can be completely misrepresented through a wormhole. When routes are established dynamically based on the reception of a packet as in Tiny OS beaconing or directed diffusion, sinkholes are easy to create because there is no information for a defender to verify. A technique for detecting wormhole attacks is presented in, but it requires extremely tight dynamic time synchronization and is thus infeasible for most sensor networks [12]. The best solution is to carefully design routing protocols in which wormholes and sinkholes are worthless.

C. Secure communication credential control

In sensor networks, all devices communicate through a wireless channel. As a result, the information flow can be easily accessed by anyone in the vicinity, and all packets are unprotected against any kind of communication attack. Therefore, it is necessary to establish a secure communication channel between sensor nodes, where no attacker can eavesdrop, modify, replay or inject messages [13]. In order to create this channel it is necessary to use security primitives, and it

is also essential to distribute the security credentials needed by such primitives.

Security primitives- Security primitives allow the sensor nodes to give a minimal protection to the information flow, and can also be used as a foundation to create secure protocols. Those security primitives are symmetric key cryptography schemes, hash primitives, and public key cryptography. Since sensor nodes are highly constrained in terms of resources, implementing the security primitives in an efficient way (using less energy, computational time and memory space) without sacrificing the strength of their security properties is one of the most important challenges in this area.

Symmetric Key Cryptography (SKC) primitives use the same secret key to hide unhidden information through encryption and decryption. Instances of these primitives are able to provide confidentiality to a certain information flow, i.e. origin and the destination of the data share the same secret key. They can also provide integrity and authentication if a certain mode of operation is used. Cryptographic hash functions or hash primitives are utilized in order to compress a set of data of variable length into a set of bits of fixed length. The result is a "digital fingerprint" of the data, identify a hash value. A cryptographic hash function must satisfy two properties: i) given a hash value *h*, it should be hard to find a message m such that hash (m) = *h*. ii) it should be hard to find two different messages m1 and m2 such that hash (m1) = hash (m2). Hash functions are typically used to assure the integrity of the information flow; provide a unique fingerprint for every packet in the form of a Message Authentication Code.

Finally, **Public key cryptography (PKC)**, also known as asymmetric cryptography, is a form of cryptography that uses two keys: a key called secret key, which has to be kept private, and another key named public key, which is publicly known. Any operation done with the private key can only be reversed with the public key, and vice versa. This nice property makes all PKC-based algorithms useful for authentication purposes. Still, the computational cost of calculating their underlying operations had hindered its application in highly-constrained devices, such as sensor nodes.

The process by which public key and symmetric key cryptography schemes should be selected is based on the following criteria:

1) Energy: how much energy is required to execute encrypt/decryption functions
2) Program memory: the memory required to store the encryption/decryption program
3) Temporary memory: the required RAM size or number of registers required temporarily when the Encryption /decryption code is being executed
4) Execution time: the time required to execute the encryption/decryption code.
5) Program parameters memory: the required memory size to save the required number of keys used by the encryption/decryption function.

In order to create and share a secure channel by using security primitives, the sensor nodes need to share certain security credentials. For example, in SKC, when one node A encrypts the information with a certain secret key K, the other node B will need the same secret key K for obtaining the original information through decryption. The task of generating and distributing those keys has to be done by a Key Management System (KMS). There are three basic factors that every key management system for sensor networks should adequately fulfill: key storage, key distribution, and key maintenance.

- **Key storage policies** indicate the number of keys that a sensor node needs to store in order to open a secure channel with other peers.
- **Key distribution protocols** define how the keys are issued to the sensor nodes. A sensor node can receive its keys before the initial deployment of the network or create its keys after the deployment using preloaded information.
- **Key maintenance procedures** specify how a sensor node can be included into or erased from the network, receiving or nullifying a set of keys in the process.

There are two extreme design cases for a key management system: global keying and pair-wise keying. In global keying, a single key is created for the entire network, and all the secure communications must be encrypted with that key. In the other case, pair-wise keying, a node must store a key for every other node inside the network, thus every pair of nodes will have a particular secure channel. In global keying, any tampered sensor node will reveal the global secret key, thus opening all the communications of the network to any potential attacker. Also, pair-wise keying is not a scalable solution due to the memory constraints of the sensor nodes. Therefore, security researchers have been trying to develop more optimal solutions that are scalable and resilient, amongst other properties. The existent systems developed by researchers can be classified into four frameworks: Key Pool Framework, Mathematical Framework, Negotiation Framework, and Public Key Framework.

Efficient and dynamic random key distribution schemes need to be designed. Most current symmetric key schemes for WSNs aim at link layer security for one-hop communications, but not the security vulnerabilities in wireless sensor networks: A Survey transport layer security for multi hop communications, because usually, it is unlikely for each node to store a transport layer key for each of the other nodes in a network due to the huge number of nodes. A more promising approach is to combine both symmetric and asymmetric cryptography techniques. Another challenging issue is that each node needs to discover a

neighbor in wireless communication range with which it shares at least one key. A good-shared key discovery approach should not permit an attacker to know shared keys between every two nodes. For any pair of nodes that do not share a key and are connected by multiple hops, there needs to be assigned a path-key to guarantee end-to-end secure communication. Such there is need to develop a dynamic path key distribution and establishment needs to be improved [14].

IV. Strength WSN Protocols

To protect the communication channel keeps sensor networks safe against certain attacks, it does not entirely guarantee that other attacks will not affect them. For example, a DoS attack can stop the provisioning of the services, and other protocol-specific attacks crafted by malicious insiders can influence over the internal behavior of the network. Therefore, it is necessary to i) strengthen the minimal set of protocols that sensor networks need in order to function properly (i.e. its "core" protocols), and ii) create specialized protocols and services that can adequately provide support for the protection of the network. This will focus on the core protocols of the network, which are routing (transmitting a packet from one sensor node to another sensor node), data aggregation (briefing many sensor readings into one single piece of data), and time synchronizing the clocks of the network. [15]

A. Routing

To design routing algorithms is a challenging area. All the nodes inside a sensor network should be reachable (*connectivity*) while covering the maximum possible area of environment using their sensors (*coverage*), even when the nodes in-side the network start to fail due to energy issues or other problems (*fault tolerance*). The algorithm should also work with any network size and node density (*scalability*) and provide a certain quality of service. At the same time, designers must try to lower the memory usage, speed, and energy consumption of the algorithms. Security is another factor that cannot be ignored in the design of routing algorithms. Any potential adversary has a wide range of attacks at his disposition to manipulate the routing subsystem and take control over the routes, resulting in eavesdropped, altered, spoofed or discarded packets [16]. He can direct traffic over his own nodes by advertising them as nodes with better (real or fake) connectivity or speed. He can alter the routing control packets on his own benefit, and also spoof the identity of the nodes using a Sybil attack.

The key infrastructure may help in the protection against routing attacks by authenticating nodes and protecting the confidentiality and integrity of the packets, but it is not enough to protect the routing infrastructure. Any adversary can take control of a set of legitimate nodes of the network and modify or discard any control messages on his own benefit. He can also attack the network or certain sections of it using a denial of service attack, jamming the communication signal and/or crash certain control packets. Therefore, it is essential to make the routing algorithm robust against such attacks, by means of multiple braided paths, restricting the structure of the topology, and other protection mechanisms.

B. Data aggregation

Inside sensor networks, the nodes generate an immense amount of raw data product of their measurements. In most cases all these information must be sent to the base station, thus there is a great cost, in terms of energy consumption and bandwidth usage, on transporting all the data from the nodes to the base station. However, since nodes are physically near each other, there will be some data redundancy. The role of aggregation is to exploit this redundancy by collecting the data from a certain region and summarizing it into one report, hence decreasing the number of packets sent to the base station. Aggregated data can be easily attacked by a malicious adversary, even if the communications are protected against any data injection attack or data integrity attack.

If an aggregator node is being controlled by an adversary, it can easily ignore the data received from its neighbors and create a false report. Trusted aggregators can still receive false data from faulty nodes or from nodes being controlled by an adversary. By using strong aggregation functions that are resilient against internal attacks, it should be possible to defend the network against false data coming from malicious or faulty nodes. Another possible solution is to try to discover whether the reports sent by a malicious aggregator are forged or not. One approach is to query the aggregator itself about the data used to create the report. Other approaches take advantage of the density of sensor networks by using the nodes in the neighborhood of the aggregator as witnesses. Finally, it is also possible to filter the packets containing the report and the proofs in their way to the base station, hence decreasing the amount of traffic created by false aggregations.

C. Time synchronization

The major task of sensor networks is the collection of data from a certain physical environment. But the data itself should be linked to the time it was collected in order to be properly used. Although sensor nodes are able to know the local time, i.e. the time that has passed after the moment they were born, it is necessary to create a set of protocols that allow the maintenance of a global time. Such protocols are also essential for synchronizing the clocks of the sensor nodes, avoiding problems such as clock drifts. If these issues are not dealt with, services such as tracking and localization may provide erroneous results. A time synchronization protocol must comply with certain design principles. The use of high-demanding energy devices such as GPS should be limited to powerful nodes (energy efficiency),

the number of transmissions should be kept to a bare minimum (transmission efficiency), factors such as the latency error and the jitter should be taken into account (end-to-end latency), and the protocols should deal with message loss and problems in message delivery (fault tolerance). Finally, the protocol must stay secure: any protocol that incorrectly updates the clock of a single sensor node can easily thwart the behavior of the entire system.

V. Other Certain Monitoring Mechanisms

There are also certain security mechanisms and systems that can assist the decision-making processes of a sensor node. Besides, it will also show other protocols and services that also need to be protected.

A. Infringement Detection Systems

The task of infringement Detection Systems (IDS) in Wireless Sensors Networks (WSNs) is to monitor sensor network systems; detecting possible intrusions in the WSNs environment, alerting users after intrusions had been detected and reconfiguring the network if possible. An infringement can be defined as a set of actions (i.e. attacks), either external or internal to a certain system, that can lead to an unauthorized access or alteration of the system. In a WSN environment, such IDS allows sensor nodes to monitor themselves and to react to the abnormal situations in their environment, providing an infrastructure that protects their normal operations and detects and reacts to any possible attacks against network services. The infringement detection approaches can be classified into anomaly based and signature based which any network security tools are mostly using [17].

One more classification can be made by considering the source of data used for intrusion detection. The taxonomy can be given based on the information derived from a single host (named as Host based IDS (HIDS)) and the information derived from complete segment of the network that is being monitored (named as Network based IDS (NIDS). An infringement Detection System must decide carefully where to locate its detection agents, due to the distributed nature of the network and the constraints inherent to the nodes. Choosing an adequate set of lightweight detection procedures, like automata, packet analysis, and health monitoring systems, is also of importance.

B. Trusted Authentication Systems

The concept of trust belief in the reliability or truth or strength of an entity derives from sociological or psychological environments, and it can be applied to a computer environment. It helps the members of a network to deal with uncertainty about the future actions of other participants. As a result, trust becomes especially important in distributed systems such as sensor networks because it may assist the execution of other protocols and services, by using the output of a trust management system as an assistant in their decision-making process. It is possible to point out some aspects that can be considered crucial for creating a satisfactory trust system. One is the initialization of the trust model. Another is the interpretation of the events that occur during the lifetime of the network. [18] The evolution and density of the events that occur in sensor networks are also of importance. Finally, due to the constraints of the nodes, it is imperative to balance the overhead of the data collection process, and to make both these processes and the trust and reputation models as lightweight as possible.

The base station is considered by the sensor nodes as a completely trusted entity. Consequently, it is of extreme importance to assure the authentication of the packets that supposedly come from the base station. It may be possible to use PKC for signing the contents of the message, or use SKC-based techniques like TESLA in order to save resources. However, in certain cases, it may be necessary to check for the authenticity and integrity of a complete stream of data. This is the case of code update protocols, which allow the whole network to be reprogrammed from the base station (sending either interpreted code or machine code) by using the wireless channel. For this purpose, it is possible to sign and send the hashed value of the entire stream, include inside the signed i packets the hash value of i+1 packets (i.e. hash-chain), or to use one-time signatures alongside with those hash-chains.

C. Privacy

As a final note, there is a security property that is very important in certain scenarios that is privacy. For example, in a battlefield, it would be important to hide the location and identities of the base station and the nodes that generated the information. In contrast, in an earthquake rescue situation locating the source nodes is an absolute must. Sensor networks could be used as a surveillance tool to collect data about the behavior of human beings.

Other security issues that have to be taken into account while using sensor networks. For example, in case a sensor network is organized into zones using an overlay, it is necessary to offer secure methods to manage that overlay, such as secure node lookup protocols. Also, there may be some mechanisms that allow a fine-grained access control in case the users can directly query the contents of a sensor network through delegated base stations. Those delegated base stations have delegated privileges from the main base station, thus in this context is essential to organize the delegation of privileges, granting or revoking them if necessary. There are other aspects that need to be protected if included inside a sensor network, such as distributed computing, secure location, secure mobile base station location, and so on. Moreover, other aspects such as the existence of random number generators in sensor nodes should be analyzed carefully.

Table 1: Attacks on WSNs and countermeasures

Layer	Attacks	Defense
Physical	Jamming	Spread-spectrum, priority messages, lower duty cycle, region mapping, mode change
Link	Collision	Error-correction code
	Exhaustion	Rate limitation
	Unfairness	Small frames
Network	Spoofed routing information & selective forwarding	Egress filtering, authentication, monitoring
	Sinkhole	Redundancy checking
	Sybil	Authentication, monitoring, redundancy
	Wormhole	Authentication, probing
	Hello Flood	Authentication, packet leashes by using geographic and temporal info
	Ack. flooding	Authentication, bi-directional link authentication verification
Transport	Flooding De-synchronization	Client puzzles Authentication

VI. Performance Metrics of Sensor Network

It is necessary to examine a list of metrics that determine the performance of a sensor network. [13]

Energy Efficiency/System Lifetime: The sensors are battery operated, rendering energy a very scarce resource that must be wisely managed in order to extend the lifetime of the network.

Security Vulnerabilities in Wireless Sensor Networks: It is always advantageous to have the ability to deploy a network over a larger physical area. Multi-hop communication techniques can extend the coverage of the network; but increase the power consumption of the nodes, which may decrease the network lifetime. They require a minimal node density, which may increase the deployment cost.

Cost and Ease of Deployment: For system deployments to be successful, the WSNs must configure itself for any possible physical node placement. In the long term, the total cost of ownership for a system may have more to do with the maintenance cost than the initial deployment cost.

Response Time/Latency: Many sensor applications require delay-guaranteed service. Protocols must ensure that sensed data will be delivered to the user within a certain delay. The ability to have low response time conflicts with many of the techniques used to increase network lifetime.

Accuracy: Obtaining accurate information is the primary objective; accuracy can be improved through joint detection and estimation. *Fault tolerance*: Robustness to sensor and link failures must be achieved through redundancy and collaborative processing and communication.

Scalability: Because a sensor network may contain thousands of nodes, scalability is a critical factor that guarantees that the network performance does not significantly degrade as the network size increases. Transport capacity/throughput: Because most sensor data must be delivered to a single base station or fusion center, a critical area in the sensor network exists, whose sensor nodes must relay the data generated by virtually all nodes in the network. Apparently, this area has a paramount influence on system lifetime, packet end-to-end delay, and scalability.

Security: WSNs must be capable of keeping the information they are collecting private from eavesdropping. Use of encryption and cryptographic authentication costs both power and network bandwidth. This impacts application performance by decreasing the number of samples than can be extracted from a given network and the expected network lifetime.

Self-healing: sensors may fail or run out of energy. The remaining sensors may need to be reorganized to maintain a set level of security.

Flexibility: key management needs to be flexible so as to allow for different network deployment methods, such as random node scattering and predetermined node placement.

Assurance: assurance is the ability to disseminate different information at different levels to end-users. A security scheme should offer choices with regard to desired reliability, latency, and so on.

Analyze security and survivability requirements concern with the design goals of scalability, efficiency, key connectivity, resilience and reliability. Security services include the following**:** *Authentication* ensures that the other end of a connection or the originator of a packet is the node that is claimed. *Access-control* prevents unauthorized access to a resource. *Confidentiality* protects overall content or a field in a message. Confidentiality can also be required to prevent an adversary from undertaking traffic analysis. *Privacy* prevents adversaries from obtaining information that may have private content. The private information may be obtained through the analysis of traffic patterns, i.e. frequency, source node, routes, etc. Ensures that a packet is not modified during transmission is known as *Integrity. Authorization:* authorizes another node to update information (import authorization) or to receive information (export authorization). *Anonymity* hides the source of a packet or frame. It is a service that can help with data confidentiality and privacy.

Non-Repudiation proves the source of a packet. In authentication the source proves its identity. Non-repudiation prevents the source from denying that it sent a packet. *Freshness* ensures that a malicious node does not resend previously captured packets. *Availability* mainly targets DoS attacks and is the ability to sustain the networking functionalities without any interruption due to security threats [2]. *Resilience to*

attacks required to sustain the network functionalities when a portion of nodes is compromised or destroyed. In *Forward secrecy* a sensor should not be able to read any future messages after it leaves the network. In *Backward secrecy* a joining sensor should not be able to read any previously transmitted message. *Survivability* is the ability to provide a minimum level of service in the presence of power loss, failures or attacks. Ability to change security level as resource availability changes is the *degradation of security services.*

Each protocol layer notably emphasizes different aspects for the security provisioning in WSNs. The physical layer provides information privacy using encoding. The link and network layers deal with the encryption of sensitive data and routing information. The application layer, higher layer in the protocol stack focuses on key management mechanism and rekeying, which in turn supports encryption and decryption of the lower layers. When considering the security issue of sensor networks, we must be aware of the characteristics of each layer, then construct a cross-layer approach to trade security off network performance and alleviate as much redundancy as possible.

Need to justify the choice of a security transversal layer by analyzing the interaction between security services. Certain security services (e.g. code attestation) might be used only by one layer (e.g. application layer), thus it is possible to think that those services should be integrated within the layer that uses them. However, those security services will probably interact with other security services (e.g. infringement detection systems) which have interactions with many layers. Therefore it is more adequate to locate the security services inside the transversal layer. The use of a transversal layer allows the modification of security services and the inclusion of new security services, limiting the possible collateral effects. [18]

VII. Conclusion & Future Work

The holistic approach of security concerns about involving all the layers for ensuring overall security in a network. For such a network, a single security solution for a single layer might not be an efficient solution rather security is to be ensured for all the layers of the protocol stack could be the best option. Cross-layer architectures the security layer behaves as another component of the architecture that can be accessed from any other component, providing security-related services and information. By centralizing all security services inside one single component, improve the overall stability and maintainability of the architecture, and also provide a better control over the possible interactions and dependencies that may arise.

Various constraints in WSN the following aspects should be carefully considered when designing a security scheme: Power efficiency, Node Density and Reliability, Adaptive security, Self configurability, Simplicity and local ID. Moreover, a highly secure mechanism inevitably often consumes a rather large amount of system resources, which in turn may unintentionally cause a security service Denial of Service attack. The outcome of this research confirm that stimulating the cross-layer design interfacing with WSN protocols will definitely improve the security services of the complete wireless sensor network efficiently. As a result, the cross-layer design is believed to provide a better security solution for wireless networks.

References

[1] Liang Song and Dimitrios Hatzinakos. A cross-layer architecture of wireless sensor networks for target tracking. IEEE/ACM Trans. Netw., 15(1):145–158, 2007.

[2] E. Shi, A. Perrig, Designing secure sensor networks, IEEE Wireless Communications (2004)

[3] Healy.M, Newe.T, Lewis.E, "Security for Wireless Sensor Networks: A Review", IEEE Sensor Application Symposium, New Orleans, LA, USA-Feb 17-19, 2009.

[4] Idrees SK, Chee-Onn C, Hiroshi I, Tanveer AZ (2010). Threat Models and Security Issues in Wireless Sensor Networks "proc. (ICINC 2010), 1: 384-389. Kuala Lumpur.

[5] Raymond, D.R, Midkiff, S.F, „Denial of Service in Wireless Networks: Attacks and Defences, IEEE CS: Security and Privacy, 2008,pg 74-81.

[6] Y. Huang, W. Fan, W. Lee and P. Yu, "Cross-feature analysis for detecting ad-hoc routing anomalies," in Proceedings of the 23rd International Conference on Distributed Computing Systems, 2002.

[7] M. Pugliese, "Managing Security Issues in Advanced Applications of Wireless Sensor Networks," PhD Thesis, University of L'Aquila, 2008

[8] M. Pugliese, A. Giani, and F. Santucci, "A Weak Process Approach to Anomaly Detection in Wireless Sensor Networks," In Proc. 1st Intern. Workshop on Sensor Networks (SN08), Virgin Islands, 2008

[9] Perrig, R. Szewczyk, V. Wen, D. Culler, J.D. Tygar, SPINS: security protocols for sensor networks, in: Proceedings of MobiCom '01, July 2001, pp. 189–199.

[10] Mingbo Xiao, Xudong Wang, Guangsong Yang," Cross-Layer Design for the Security of Wireless Sensor Networks", Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China, pp(104-108).

[11] Kalpana S, Ghose (2011). Cross Layer Security Framework for Wireless Sensor Networks. IJSA, 5(1): 39

[12] M. C. Vuran and I. F. Akyildiz, "Cross-Layer Analysis of Error Control in Wireless Sensor Networks," in Proc. IEEE SECON '06, Reston, VA, September 25-28, 2006.

[13] Slijepcevic S, Potkonjak M (2002). On Communication Security in Wireless Ad-Hoc Sensor Networks. Eleventh IEEE International WETICE, 1(1): 139-144.

[14] Ergun, M. ; Levi, A. ; Savas, E., "A resilient key pre-distribution scheme for multiphase wireless sensor networks", Computer and Information Sciences, 2009.

[15] M Healy, T Newe and E Lewis,"Resources Implications for Data Security in Wireless Sensor Network Nodes" in Sensor Comm 2007,pp(170-175).

[16] C. Karlof, D. Wagner, Secure routing in sensor networks: attacks and countermeasures, in: Proceedings of the 1st IEEE Workshop on Sensor Network Protocols and Applications, May 2003, pp. 1–15.

[17] Ozgur, D., T. Murat, et al. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Systems and Applications 29(4):pp. 713-722

[18] L. van Hoesel, T. Nieberg, J. Wu, and P. J. M. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," IEEE Wireless Communications, vol. 11, no. 6, 78 - 86, Dec. 2004.

Prof. Sanjeev Puri: Research scholar and pursued PhD (CS &E) from Singhania University, Jhunjhunu. He is the Reviewer editorial member of IACSIT-IJCEE, IEEE-ICMLC and Elsevier. He is working as Professor at SRMGPC (Now SRM University), Lucknow, India. His research interests in wireless sensor networks security, grid security and protocols

Dr. S.P. Tripathi: has done his PhD (Computer Sc. & Engg) from Lucknow University. He is working as Professor (CS & E) at IET, Deemed University, Lucknow. His research interests in Data and Information security.

