Security in Wireless Sensor Network: A Current Look

Mohammad Luqman
Department of Computer Science
Aligarh Muslim University
Aligarh, India
luqman.geeky@gmail.com

Arman Rasool Faridi
Department of Computer Science
Aligarh Muslim University
Aligarh, India
ar.faridi.cs@gmail.com

***Abstract*—Wireless Sensor Network (WSN) is used to enhance and automate the information capturing capabilities of critical projects such as Medical Monitoring, Scientific Experiments and Research, Industries, Factories, Transportation, etc. Wearable devices, Smart Cities, Smart Agriculture, Virtual Reality (VR), Augmented Reality (AR) gaming are some of its newer applications. Governments also use WSN to gather data for their military applications. WSN nodes have gained this massive adoption because they are cheap, modular, programmable, ad-hoc configurable, can operate in an open environment, wirelessly connected, etc. These features also make WSN easy to attack. Data captured by WSN are private or sensitive for individuals or organizations. Hence, security is one of the essential issues of research of WSN. In this paper, we present a current brief review of WSN Security. First, WSN and its architecture is introduced. Then, a layer-wise classification of security attacks and possible mitigation techniques are discussed to understand attacks and their severity. To secure WSN, lightweight cryptographic algorithms are analyzed for their suitability, performance, battery requirements, etc.**

Keywords—Wireless Sensor Network, Security, Lightweight Cryptographic Algorithms, Lightweight Block Cipher, Lightweight Stream Cipher, Lightweight Hash Algorithm.

I. Introduction

Wireless Sensor Network (WSN) has a decade's old origin [1]. Like any computer technology, the origin of WSN is attributed to military requirements and heavy industrial applications. The history of WSN is much older than the Internet and even from most generations of computers. The first system that can mimic modern WSN was Sound Surveillance System (SOSUS) [2] developed in the 1950s by US Military. SOSUS was used to track and detect Soviet submarines. Further, with the establishment of the Defence Advanced Research Projects Agency (DARPA) project by the US Govt. Distributed Sensor Network (DSN) program was formally started in 1980. After its formal introduction, research interests started to rise and people from academics and research labs showed huge interest. Later DSN was modified into modern WSN. WSN in the most basic term, is a network of sensors spread over small or large geographical areas that are used to sense and collect any type of physical or environmental quantity like heat, temperature, light, sound, moisture, suspended particle matter etc.. The data collected via these sensors are transmitted to a central device, which acts as a sink. Central device or sink further transmit these data to its destination over the network. Sensors are cheap and easy to install, leading to WSN's huge adoption in its early years. Sensors communicate with each other using radio waves. Most WSN networks are Ad-Hoc networks in which sensors are scattered over the particular geographical area, and the nodes are self-configured such that they communicate with each other and determine the best path to reach the sink by themselves using strategies like lowest node count to sink, a path with high battery power, a path with powerful nodes etc. A WSN node comprises of central processing unit, a communication unit, a sensing unit, and a power unit. Nodes are low-powered (battery or energy-source), low-power computation devices whose only purpose is to sense and collect data and transmit it to the sink. Since the nodes are not statically routed, constantly monitored or secured, they are susceptible to a large array of attacks.

Our objective in the paper is to present a contemporary look at the WSN network and how nodes in a WSN network are attacked, and what mitigation techniques are used to prevent those attacks. To fulfil these objectives, this paper takes the structured approach, which are:

- A short description of the WSN network's structure is discussed to explain the various layers with their specific roles and responsibilities.
- Potential security issues that are created due to the distributed or Ad-Hoc nature of WSN [3].
- To simplify the security attacks on WSN, it is classified into the layers the attacks operate upon [4].
- A brief of lightweight Cryptographic Algorithms that is suitable to use in WSN.
- Some of the most common Lightweight Cryptographic Algorithms used as well as some recently developed ones are presented that are used in practical scenarios.
- Current research issues that are undergoing or need further research is presented.

II. Architecture of WSN

Network stacks are broken down into layers to decouple the complexity and responsibilities into different layers as well as to allow easier debugging and maintenance of the stack. The most notable of network stacks are OSI model and TCP/IP model. WSN network stack is also divided into five-layer architecture [5] similar to OSI model which is presented in Figure 1.

978-93-80544-44-1/22/$31.00 ©2022 IEEE

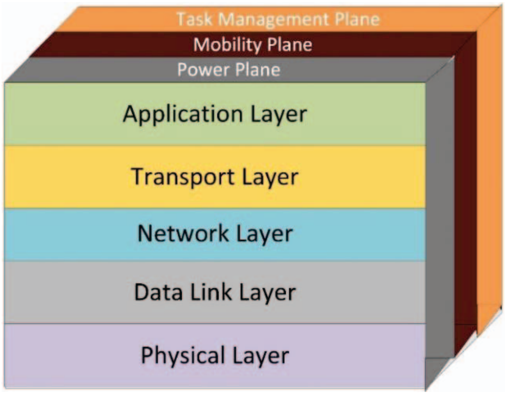

Fig. 1. Layered Architecture of WSN

- Physical layer in WSN is similar OSI model and lies at the bottom of the network stack. The layer establishes the link to communicate between motes and receive/transmit data in binary format. This layer performs tasks such as searching suitable frequency range to transmit data, generating carrier frequency for data transmission, deflecting signals, signal modulation, data encryption and receiving of data signals.
- Data Link Layer is responsible for various functions such as multiplexing of data signals, collision detection, error control, detection of data frames, control access to a transmission medium (MAC) and providing reliable point-to-point and point-to-multipoint communication.
- Network Layer performs tasks such as allocation of network addresses which are based on data or location and not as unique number I.D.'s since the size of WSN network can be very large, internetworking with external networks and to manage the forwarding of data packets.
- Transport Layer is required to provide information to achieve the reliable transport of data packets. Routing of data packets by the network layer is performed by the information passed by the transport layer. Although end-to-end reliability of data transmission cannot be guaranteed like the guarantee of reliability by transport layer in OSI model due to the lack of unique global addressing scheme, power dissipation, scalability of the network etc. But WSN Transport layer can guarantee localized reliability.
- Application Layer is the topmost layer in the WSN stack. The layer controls how the data is structured and formatted before it is sent to other nodes in the network and specifies the format and structure of data which can be received from other nodes. The layer abstracts all the other layers in WSN and acts as a programmable unit, allowing the execution of various applications and providing user interaction with it.

Splitting the WSN network stack into these five layers allows to specify the roles and functions to different layers and hence reduces overall complexity in developing and maintaining the network stack. But WSN nodes are also required to self-manage its computation cycle, energy consumption and data collection/dissipation rules. Hence WSN embodies three planes which are shown across the network stack as it involves assistance from multiple layers which are: Power Plane is responsible to monitor and control the power usage of the nodes. To save power, nodes often go into sleep mode. The plane can instruct node to go into sleep mode after receiving data so that duplication of data reception is prevented and also to save power. If a node has low power left, it can instruct its neighbours via broadcast messages that it will stop participating in future routing updates as it has low power left and the remaining power will only be used for sensing. Mobility Plane monitors the node's location in the network and detects any movement in the node. If there is any movement or change in location, it will modify the neighbour location of the node and the energy dissipation required by the node to route data to other nodes in the network. The plane monitors these so that the node can relay accurate information to other nodes to achieve reliable routing of data. Task Management Plane manages the distribution of tasks among the nodes. Since nodes may be distributed over vast geographical areas into various regions, there might come some tasks which are not required to be performed by all the nodes in the network, but a small section of nodes in the network lying in that specific region. The plane is required to achieve the optimum usage of the WSN network by creating power-efficient ways to perform tasks, routing data reliably in the network, and sharing the network's resources.

These planes work in conjunction with other layers as well as with other nodes in the network by balancing the usage of WSN network. The planes carefully allocate the tasks and monitor the power usage of all the nodes in the WSN network. The balance in the network is required so that the network can perform maximum tasks in its lifetime otherwise working of individual node discreetly will produce reduced result and lowers overall lower performance of the network similar to the Nash equilibrium.

III. Security Issues of WSN

WSN is different from other networks like LAN, WAN, Cellular network etc. WSN networks are mostly Ad-Hoc and comprise of hundreds to thousands of sensor nodes spread densely in a geographical area [6]. The whole network works as an entire unit and provides a data-centric approach to its user. However, WSN's features that provide excellent facilities to its users also open wide areas of security issues [7], [8]. Some security issues are similar to other networks and some have an even more significant threat than wired networks. The features of WSN are as follows:

A. *Self-Configuration/Organization*

WSN does not maintain or create any network topology. Its structure, size and position are all created and maintained by the nodes themselves. Any failure of node or path is to be mitigated by the nodes themselves via constant and periodic route updates.

B. *Flow Control*

Routes in a WSN network are created based on the energy requirement of the path and the power level of the participating nodes. Any error in the flow of data packets in the network should be automatically handled by the nodes via route update messages so that performance of the network can be maintained.

C. Limited Resources

WSN nodes are battery-powered (or limited power with a rechargeable source like solar, tidal, etc.), low computationally capable sensor devices. They can't manage complex and high computationally intensive algorithms and tasks. So, it is required to create appropriate power-efficient lightweight cryptographic algorithms to secure the device. The lightweight cryptographic algorithms can prevent most external attacks, but they can't prevent attacks from inside.

D. Open Environment

Nodes in WSN are placed in an open environment, allowing attackers with different ranges of attacks that can be performed internally and externally. No constant monitoring and dynamic route updates are features of an open environment that should be secured to prevent any complete takeover of the network by an attacker using any internal or external attack.

E. Mobility of nodes

Nodes in WSN can change their location in the network and inform other nodes in the network by route update messages. Mobility of nodes can also allow black hole and sinkhole attacks which will drain overall power consumption of the network and cause low efficiency.

F. Heterogeneity of Nodes

It implies nodes that differ in terms of power, computational capacity, memory, etc. There is a lot of focus in researching routing protocols for heterogeneous WSN. Heterogeneous WSN can provide more optimized and efficient network usage than homogenous WSNs. But it can also create security issues e.g., if malicious nodes start advertising incorrect information of themselves and try to take down the network; there is a difference between uplink and down-link; or an incorrect node is selected for transmission in a region.

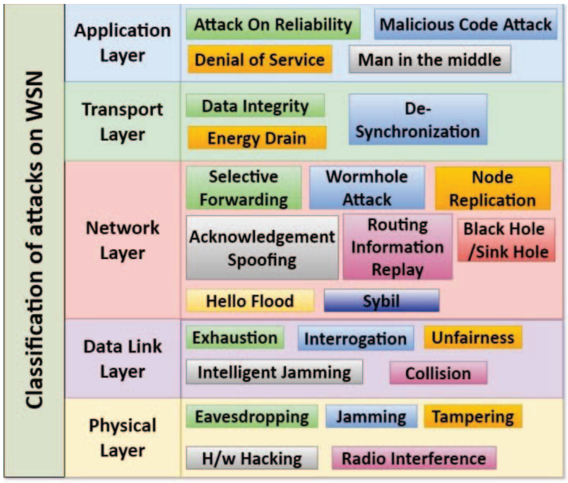

Fig. 2. Classification of Attacks on WSN

G. Centralized System

Nodes in WSN is generally scattered across a particular geographical area but they send their data to the sink device instead of sending it by themselves due to battery considerations, lightweight computation limitations. The sink device provides the centralized gateway of transmission of sensor data and node health information to its recipients. Its failure can lead to fatal network failure or data shut down for the entire network.

IV. Layer-wise Classification of Attacks on WSN

A layer is the basic unit of operation in WSN. To get a simplistic view, the following sub-sections briefly describes the nature of attacks categorized under different layers they acts upon [9], [10] (Figure 2).

A. Physical Layer

Eavesdropping is an external attack. The attacker quietly listens and monitors the traffic between nodes and then extracts useful information from it. It is hard to detect whether the communication is being eavesdropped on due to its passive and silent nature. So, it is required to secure the communication from being eavesdropped by encrypting the message. Even if the attacker captures encrypted messages, he/she cannot extract any information because the message is illegible. *Jamming* is an external attack on WSN nodes. In this, the attacker produces noise signals similar to signals used for communication by nodes but with high power. It is intended to disrupt the on-going communication inside the network to bring down the network. If the noise signals have sufficient strength it can disrupt the whole or a part of the network. *Tampering* is an internal/external attack both on the WSN node itself. In this, an attacker extracts useful information from physical layer on WSN node itself such as cryptographic keys or any sensitive data. The attack can be prevented by tamper-proofing the WSN node physical unit, but it is rarely done as it increases the overall cost. *Hardware hacking* is an internal/external attack both on the WSN node itself. In this, an attacker defaces or removes some part of hardware from the WSN node. The WSN node loses some functionality that can be further exploited to extract useful information. The attack can also be prevented by tamper-proofing the WSN node physical unit. *Radio interference* is an internal attack. In this attack, a malicious node produces irregular interferences to block out or tamper with outgoing communication from its neighbour.

B. Data Link Layer

Collision is an internal attack. Two nodes try to transmit messages on the same frequency as the other. When the frequency of these signals collides, they create a tampered value. The tampered value will be detected by the checksum calculated and the message will be discarded. An attacker can maliciously block important messages to lower the network efficiency such as creating collisions with ACK packets which force nodes to create exponentially time costing back-off in some MAC protocols. Collision can be avoided by implementing error-correcting codes, but it will add an overhead and time cost for the packet. Also, error-correcting codes can't recover from highly damaged packets as intended by an attacker. *Intelligent jamming* is an external attack. In this, an attacker creates specially crafted messages intended to target a MAC protocol's weakness or fallback. These malicious packets will cause the nodes to lose their energy very fast, create congestion in the WSN network or will destroy the entire communication signal. *Unfairness* is an internal attack. It uses other Data Link attacks to degrade the services provided by the network. By worthlessly occupying the network service in order to process these useless packets, it will hinder real-time services of the network to valid nodes. It can be prevented by using small frames to send messages but that will also lower the overall network efficiency and can't guarantee to prevent useless packets with high frequency. *Interrogation* is an internal/external attack both. It aims to exploit the two-way handshaking protocols implemented in various MAC protocols. Two-way handshaking protocols are used to mitigate the hidden node problem in WSN network. In this, an attacker will repeatedly send the request-to-send (RTS) message to obtain a valid clear-to-send (CTS) message from intended targets. Consuming RTS messages and producing CTS messages drains the power of nodes. It can be prevented by employing Anti-Replay mechanisms and a strong MAC layer authentication mechanism, but the node still has to lose energy to process the RTS message. *Exhaustion* is an internal/external attack both. In this, an attacker uses repeated collisions to exhaust the resources of a node and its neighbour. A trivial MAC protocol will try to send messages repeatedly unless the corrupted packet is dropped or discovered, thus losing its own energy and its neighbours. It can be prevented by rate-limiting the admission packets to ignore such packets after a fixed number of tries. Another solution is to use Time-Division multiplexing where frames are allotted time slots to deliver the message.

C. Network Layer

Selective forwarding is an internal attack. A naïve implementation will assume that all nodes will transmit all the packets it receives to the network. In this attack, an attacker will insert a malicious node in the network which will drop some packets and forward some packets according to some predefined rules. It can be prevented by using multipath transmission or detecting the malicious node and alerts other nodes not to use that node for transmission. *Routing information replay* is an internal/external attack both. Routing information is the message according to which the WSN network is set up. In this attack, the routing information exchanged between devices is spoofed, modified, or destroyed to alter the setup of the network. The alterations include creating routing loops, changing the path's weight between nodes, network partitioning, fake error messages, etc. It can be prevented by adding message authentication code (MAC) after each message which can be used to verify whether the message has been tampered. Replay can be prevented by using timestamps, number counters etc. *Sybil* is an internal attack. In this, attacker presents more than one identity (ID) to the network. It can be used to fool MAC protocols, distributed storage and fault-tolerant algorithms. MAC protocols used to maintain network topology can assume the presence of an extra node that doesn't exist. Distributed storage can use the non-existent Sybil node for storing some crucial information. *Black hole (sink hole)* is an internal/external attack both. Black Hole/Sink Hole are two different attacks but have many similarities. In the Black Hole attack, attackers insert malicious nodes whose only function is to drop all the packets it receives, disrupting the network. In Sink Hole attack, the attacker advertises its malicious node as a node with the lowest weight to attract most of the network's traffic. Now when the malicious node gets most of the network's traffic, it can be further used to employ other attacks such as selective forwarding, black hole etc. *Wormhole attack* is an internal attack. In this, a malicious node advertises that it has the lowest latency with other nodes in distant parts of the network or between two adjacent nodes. It is similar to sinkhole attack by attracting high traffic from its neighbours, but it differs due to high distance from distant nodes or falsely claiming low latency between adjoining nodes. *Hello flood* is an internal/external attack both [11]. Most MAC protocols understand HELLO messages as an assurance that another node is in their close vicinity. But a malicious node with a high power transmitter can deceive a large number of nodes into believing that the malicious node is in its close vicinity when in reality, it lies far away. This results in nodes trying to send their messages through the malicious node which consumes their energy and lowers the network throughput. *Acknowledgement spoofing* is an internal attack. In this, acknowledgments used in routing information are exploited to disrupt network services. Malicious nodes can generate fake acknowledgments of the captured packets to falsify the routing information such as sending acknowledgments of dead nodes to assert that they are live. *Node replication* is an internal/external attack both [12]. In this, actual nodes are captured and information like routing information, encryption keys and other sensitive information are captured. Captured node can also be used for eavesdropping.

D. Transport Layer

Data Integrity is an internal attack. Data is modified to change its payload or to insert wrong routing information. Thus bringing the whole network down with it. **De-Synchronization** is an internal/external attack both. It implies breaking down existing connections. An attacker can create a spoofed message which requires more retransmission from the end hosts. These spoofed messages are repeatedly requested to degrade or deny the services provided by the end hosts by wasting energy/computation time on useless retransmissions. It can be prevented by requiring reliable authentication from all hosts that communicate. **Energy Drain** is an internal/external attack both. It implies sending a huge number of connection establishment requests to the target node to disrupt its services. When this attack is done on the scale, it can cause a denial of service of the whole network.

E. Application Layer

Attack on reliability is an internal/external attack both. In this, attacker insert malicious nodes which alters every message that it receives, mainly the routing information. It causes collisions and useless retransmission which decreases the energy and throughput of the network. *Malicious code attack* is an internal/external attack both. It inserts worm in the application layer which takes over the node by executing a malicious program. *Denial of service* is an internal/external attack both. This attack is applicable to the application layer and to all the layers in the WSN stack. It uses any combination of the above attacks to disrupt the network by constantly attacking nodes. *Man in the middle* is an internal/external attack both. Information is retrieved during handshaking or key exchange to extract sensitive information.

V. Lightweight Cryptographic Algorithms

Cryptographic algorithms provides the mathematical proof of secure message delivery between two ends. These algorithms convert plain messages (plaintext) into garbled messages (ciphertext) so that the ciphertext can't give out any statistical inference or clue to reverse into meaningful context. Cryptographic algorithms are classified majorly into three parts:

- Encryption/Decryption algorithms are used to convert plaintext into ciphertext and vice-versa. These algorithms provide the confidentiality goal of a secure system since messages are converted into meaningless forms and apart from sender and receiver no other person extracts any meaningful information from it.
- Hashes/Message Authentication Code (MAC) creates cryptographically secure hashes or checksums that can be used to protect the integrity of a given data. These algorithms are one-way, which means you can repeatedly generate a hash from the message and hashes generated will always be the same. But you can't generate or predict message from the hash. MAC is generally created using a keyed hash function or block ciphers [13]. NIST has published FIPS 198-1 [9] document to specify good MAC algorithms.
- Cryptographically secure pseudorandom number generator are algorithms which produce random numbers that are cryptographically secure which means it is very hard to predict the next number generated by the algorithm even if we know the source code of the algorithm or the previous generated number in the series. Random numbers are highly critical for providing security in modern applications and systems.

This paper only briefly introduce lightweight encryption decryption and MAC algorithm since WSN nodes are limited in terms of running multi-cycle cryptographic tasks or availability of regular power supply to support the computation. Hence, lightweight algorithms should be used which provide security using the least number of resources. A good lightweight algorithm should possess the following attributes:

- Security: The security of a cipher implies the resistance to any known statistical(linear or differential) attack or algebraic attack on it [10]. It is equal to the key size used for symmetric encryption, while for asymmetric cryptography, it is half of the key size used.
- Chip Area/Gate Equivalence(GE): The design of chips also has an impact on lightweight ciphers. A fewer number of gates can lower the cost of chip production and energy consumption. The number of G.E. less than 3500GE should be considered a lightweight cipher and even GE is less than 1000, it is assumed to be a good lightweight cipher.
- Throughput: Number of bits generated during encryption decryption process in one second at a specified frequency. Higher throughput is better for it will consume less time and energy to secure the data.
- Latency: Latency implies the initial delay in generating a block of plaintext or ciphertext. Real-time applications of WSN like WBAN, VANET, Automated vehicles etc., can significantly affect with high latency. So, algorithms with lower latency are preferred.
- Figure of Merit: This metric is used to create a neutral benchmarking metric that is not impacted by unsuitable efficiency metrics generated from hardware implementation. It includes the power consumption of the algorithm and is independent of the process involved in algorithm deployment [11].

Lightweight cryptographic algorithms are optimized for microcontrollers used in popular sensor nodes such as MSP430, ATMega128 and ARM Cortex-M3 microcontrollers [12]. Lightweight cryptographic algorithms are classified into three types: lightweight block ciphers, lightweight stream ciphers, and lightweight hash algorithms and HMACs.

A. Lightweight Block Ciphers

Like their conventional counterparts, lightweight block ciphers operate on blocks of fixed size and use fixed size keys to generate cipher text but tend to consume less cycle. Table I describes some of the popular lightweight block ciphers [14] currently used with some latest alternatives.

TABLE I. Lightweight Block Ciphers

Cipher Name	**Year**	**Structure**	**Round**	**Key Size (bits)**	**Block Size (bits)**
RC5	1994	Feistel	0– 255	0–2040	32/64/128
TEA	1994	Feistel	64	128	32/64
XTEA	1997	Feistel	64	128	64
DESL	2007	Feistel	16	56	64
PRESENT	2007	SPN	31	80/128	64
CLEFIA	2007	Feistel	2488	128,192,256	39/128
KATAN KTANTAN	2009 2009	NLFSR NLFSR	254 254	80 80	32/48/64 32/48/64
MIBS	2009	Feistel with SPN round function	32	64/80	64
LED	2011	SPN	8 for 64,12 for others	64/80/96/128	64
TWINE	2011	GFN Feistel	32	80/128	64
KLEIN	2012	SPN	12/16/20	64/80/96	64
PRINCE	2012	SPN	11	128	64
ITUBEE	2013	Feistel	20	80	80
SIMON and SPECK	2013	Feistel ARX	32-72 22-34	64-256 64-256	32-128 32-128
RECTANGLE	2014	SPN	25	80/128	64
Khudra	2014	Feistel	18	80	64
Midori	2015	SPN	16/20	64/128	64/128
VH	2015	SPN	10-14	64-128	64
MANTIS	2016	SPN	10/14	128	64
HIGHT	2016	Feistel	32	128	64
QTL	2016	Feistel	16/20	64/128	64
ANU	2016	Feistel	25	80/128	64
CHAM	2017	Feistel+ARX	16-32	64-128	128-256
SFN	2018	Feistel+SPN	32	96	64
CRAFT	2019	SPN	32	128	64
LRBC	2020	Feistel+SPN	24	16	16

B. Lightweight Stream Ciphers

Lightweight stream ciphers like their conventional counterpart encrypt and decrypt blocks of variable (r) size. Table II describes some of the popular lightweight block ciphers currently used and some latest alternatives.

TABLE II. LIGHTWEIGHT STREAM CIPHERS

Cipher Name	Year	Key Size (bits)	Area (GE)	IV	Type
A5/1	1987	64	923	22	LFSR
Rabbit	2003	128	3800 4100	64	Chaotic Table + simple arithmetic
Grain	2005	80, 128	1294 3239	64, 96	LFSR, NFSR
Trivium	2005	80	2580 4921	80	3SHR
Salsa 20/r	2005	128, 256	12126	128	ARX
Grain128a	2006	128	1857 4617	96	LFSR + NLFSR
Sosemanuk	2008	128, 256	4100 2700, 18, 819	64, 128	LFSR + FSM
MICKEY	2008	80, 128	3188 5039	0-80, 0-128	Galois LFSR + NLFSR
CHACHA	2008	256	750	128	ARX
Encoro 128 Encoro80	2009 2008	128 80	4100 2700	64 64	PRNG
SNOW-3G	2010	128		128	LFSR + FSM
A2U2	2011	56	500 284		LFSR + 2NLFSR
Quavium	2012	80	3496 2372 (3 round version)	80	4 Trivium like SHR
WG-8	2013	80	1786 3942	80	LFSR + WG
Sprout	2015		813 839		NLFSR + LFSR + Counter Reg
Fruit-v2	2016	64/80	990	64	LFSR + NLFSR
Plantlet	2016	80	228	90	LFSR + NLFSR + Counter
Espresso	2017	128	1500	96	Galois structure NLFSR
Lizard	2017	120	1161	64 3578	NLFSR
ESSENCE	2019	128, 192, 256	-	-	-

C. Lightweight Hash Algorithms and MAC

Lightweight hash algorithms are generally used to provide authentication [15]. Hash algorithms when used within the encryption algorithms itself is called as Authenticated Encryption (AE). AE cryptographic algorithms provide integrity, confidentiality and authenticity. A modified version of A.E. is Authenticated Encryption with Associated Data (AEAD). AEAD algorithms can assure the integrity of both plaintext and ciphertext to its receiver. Table III mentions some lightweight hash algorithms.

VI. RESEARCH ISSUES

Much has been done in past decades in securing WSN. But as the new applications are created, securing these applications requires further research.

- Smart cities [16], Wireless Body Area Networks, Smart healthcare, Smart Homes etc. will generate data that will be very private and sensitive. Securing the data should be the utmost focus otherwise, they can create a more significant threat to its victim [17].
- Development of a security framework that is resilient to multiple attacks and harsh networking conditions and secures every aspect of WSN should be done.
- Military applications will also require silent, resilient, and secure network setup and transmission.
- Internet of Everything is the future. It will add billions of devices that will create huge infrastructure issues.
- Secure Data aggregation, efficient energy-aware routing, and Data warehousing will be another issue for the future.
- Use cases of Machine learning/Artificial Intelligence that can prevent misuse, attacks, wrong configuration of nodes or can generate route recommendations like other recommendation systems [18] need to be explored.
- WSN will also be required to collaborate with other technologies dealing with intelligence, efficient routing, secure and energy-efficient channel, etc., apart from the applications they serve themselves.
- Softwarization of WSN will allow better Network management, optimal routing path and better prevention from DoS attacks similar to SDN-IoT development [19].

TABLE III. LIGHTWEIGHT HASH ALGORITHMS AND MAC

Hash Function	Year	Size (bits)	Gate (GE)	Security
Quark	2010	128–224	1379/2296	64/112-bit
SPONGENT	2011	88-256	738-1950	80-240bit preimage security
PHOTON	2011	80–256	1120	64-bit
Neiva	2011	256	824	-
ARMADILLO	2012	48	2923	80bit
Serial Keccak-f [200]	2015	64	2.52K	64-bit
Hash-One	2016	160	1006	80-bit security/160 bit preimage resistance/80 bit collision resistance
GLUON-64	2016	128	2071	64-bit

VII. CONCLUSION

WSN networks are ubiquitous now. They are assisting future technologies to acquire reliant and fault-tolerant abilities through sensing. For now, WSN is assisting IoT in performing non-intelligent tasks. Newer applications of WSN are rising every day and security frameworks need to acknowledge that. We can easily automate most of our lives by leaving redundant tasks to machine, but there must be checks and balances to assure that these systems can't be compromised to leak important or classified data. Security is a big part of WSN, but a holistic architecture or solution is not developed yet. The study aims to assist future researchers in ascertaining the work done already and future issues to deal with. We tried to deliver the latest innovations and holistic view of WSN in terms of its security, architecture and future research directions.

REFERENCES

[1] A. Ali, Y. Ming, S. Chakraborty, and S. Iram, "A Comprehensive Survey on Real-Time Applications of WSN," Futur. Internet, vol. 9, no. 4, p. 77, Nov. 2017.

[2] M. Kocakulak and I. Butun, "An overview of Wireless Sensor Networks towards internet of things," in Proc. of the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), 2017, pp. 1–6.

[3] G. Sharma, S. Bala, and A. K. Verma, "Security Frameworks for Wireless Sensor Networks-Review," Procedia Technol., vol. 6, pp. 978–987, 2012.

[4] I. Tomić and J. A. McCann, "A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols," IEEE Internet Things J., vol. 4, no. 6, pp. 1910–1923, Dec. 2017.

[5] M. M. Bokare and M. A. Ralegaonkar, "Wireless sensor network," Int. J. Comput. Eng. Sci., vol. 2, no. 3, 2012.

[6] J. Li, L. Andrew, C. Foh, M. Zukerman, and H.-H. Chen, "Connectivity, Coverage and Placement in Wireless Sensor Networks," Sensors, vol. 9, no. 10, pp. 7664–7693, Sep. 2009.

[7] D. A. P. Kumari and M. Indu, "SENSOR NETWORK SECURITY," Int. J. Technol. Res. Eng., vol. 7, no. 9, pp. 43–68, 2020.

[8] K. CHELLI, "Security Issues in Wireless Sensor Networks: Attacks and Countermeasures," in Proc. of the World Congress on Engineering, 2015.

[9] N. FIPS-198-1, "The Keyed-Hash Message Authentication Code," Fed. Inf. Process. Stand. Publ., vol. 198, no. July, pp. 1–20, 2008.

[10] W. Julian Okello, Q. Liu, F. Ali Siddiqui, and C. Zhang, "A survey of the current state of lightweight cryptography for the Internet of things," in Proc. of the IEEE 2017 Int. Conf. Comput. Inf. Telecommun. Syst. (CITS 2017), pp. 292–296, 2017.

[11] S. Badel et al., "ARMADILLO: A multi-purpose cryptographic primitive dedicated to hardware," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2010, vol. 6225 LNCS, pp. 398–412.

[12] S. S. Dhanda, B. Singh, and P. Jindal, Lightweight Cryptography: A Solution to Secure IoT, vol. 112, no. 3. Springer US, 2020.

[13] S. Shin, M. Kim, and T. Kwon, "Experimental performance analysis of lightweight block ciphers and message authentication codes for wireless sensor networks," Res. Artic. Int. J. Distrib. Sens. Networks, vol. 13, no. 11, p. 2017, 2017.

[14] D. Sehrawat, N. S. Gill, and M. Devi, "Comparative Analysis of Lightweight Block Ciphers in IoT-Enabled Smart Environment," in Proc. of the 2019 6th Int. Conf. Signal Process. Integr. Networks (SPIN 2019), pp. 915–920, 2019.

[15] M. A. Simplicio, B. T. De Oliveira, C. B. Margi, P. S. L. M. Barreto, T. C. M. B. Carvalho, and M. Näslund, "Survey and comparison of message authentication solutions on wireless sensor networks," Ad Hoc Networks, vol. 11, no. 3. Elsevier B.V., pp. 1221–1236, 01-May-2013.

[16] Z. A. Pindar, J. O. Fayomi, N. H. Waziri, B. M. Abdulhamid, and S. Jamel, "A Lightweight Message Authentication Code for Virtual Work in Future Smart Cities," in Proc. of the 2020 IEEE Eur. Technol. Eng. Manag. Summit (E-TEMS 2020), 2020.

[17] S. Batchu, O. S. Henry, and A. A. Hakim, "A novel decentralized model for storing and sharing neuroimaging data using ethereum blockchain and the interplanetary file system," Int. J. Inf. Technol. 2021 136, vol. 13, no. 6, pp. 2145–2151, Jul. 2021.

[18] K. Anwar, J. Siddiqui, and S. S. Sohail, "Machine learning-based book recommender system : a survey and new perspectives," Int. J. Intell. Inf. Database Syst., vol. 13, no. 2/3/4, pp. 231–248, 2020.

[19] A. H. Shamsan and A. R. Faridi, "Network softwarization for IoT: A Survey," in Proc. of the 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), 2019, pp. 1163–1168.