Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhancing Data Security and Authentication in Wireless Sensor Networks with Secure Hash Based Intrusion Detection

Jeya Rani D[1], NagarajanMunusamy[2], Jeya Rani D[3]
[1] *Research Scholar, KSG College of Arts and Science, Coimbatore*
[2]*Professor and Principal, KSG College of Arts and Science, Coimbatore*
[3]*Assistant Professor, KG College of Arts and Science, Coimbatore*

Network intrusions have a significant negative impact on network security. Many network settings have made heavy use of intrusion detection technologies to address this problem. In general, intrusion detection is the process of watching computer system or network events to identify potential violations of pre-defined security regulations or conventional security practices, such as malware attackers obtaining unauthorized access to systems and human activity. Intrusion detection systems grow more resistant to a variety of attacks by automating the detection process. In this research, we enhance data security and authentication in Wireless Sensor Network with Secure Hash based Intrusion Detection (SHID) Systems. A hashing approach ensures that data is safely sent via wireless networks. In data security, these approaches utilize a secret key to encrypt the original data, into unreadable format. SHA2 and SHA3 algorithms are used to swiftly and efficiently detect threats and malwares. Data communication is authenticated using the Advanced Encryption Standard (AES) algorithm. According to the findings, the recommended strategy outperforms the existing ones.
Keywords: Encryption, Hashing, Network, Security, Wireless Sensor Network

1. Introduction

Ad hoc, independently functioning networks are wireless sensor networks (WSNs) [1-3]. The adaptability of the technology makes it possible to monitor and sense events in many spheres of applications [4-6]. It tracks physical changes in the surroundings like temperature, light intensity, radiation level, and pressure using low-cost sensor nodes with several sensing units [7-9]. Because of the cheap devices and simplicity of use, it is also often employed to monitor vast geographical areas at little expense [10-12]. Given their limited memory and processing capacity, sensor nodes must typically preserve electricity [13]. Security techniques and policies are often not designed with power-limited applications in mind, relying on memory for key storage and CPU overhead for encryption and authentication [14-

16].

Ad hoc networks are independent wireless sensor networks and the technology's adaptability allows for the detection of events in a wide range of applications [17-19]. Low-cost sensor nodes equipped with several detecting units detect ambient physical changes such as temperature, light intensity, radiation level, and pressure [20-22]. The technology's low cost and simplicity of use make it a popular choice for monitoring broad geographical areas [23-25]. Sensor nodes' limited memory and processing capability need them to save energy on a regular basis [26-28]. Security techniques and regulations are often relied on memory for key storage and CPU overhead for encryption and authentication, and they are not designed for power-limited applications [29-31].

Wireless sensors are defined by their limited storage capacity, battery life, and computational power [31-33]. Unfortunately, the nature of wireless communication in WSNs enables attackers to easily intercept conversations, impersonate other users, introduce bogus data, or alter the content of approved messages during multi-hop forwarding [34-36]. Thus, authentication methods must be employed to protect communications from certain types of hostile conduct [37-39]. Source and message authentication are similar to authentication in this regard [40]. While message authentication ensures that the data from the source is new and unmodified, source authentication ensures that the data acquired comes from the specified source [41-43]. Recent significant advancements in the design of efficient one-way hash algorithms suited for hardware and software [44-48].

2. Background Study

Al-Mashhadi, H. M., et al. (2014) these authors presented the 2 Array Message Digest (2AMD-160) hash scheme. Simultaneous usage of two AMO-160s reduced operation time and increased safety. While a little change in message size will speed up the hash function, it has been shown that the number of message blocks is critical. They compared the innovative technique to other standard methods using the same set of tests and observed that it enhanced the run time and message security of WSN nodes.

Asha, T. R., &Hamsaveni, N. (2015) Keccak provides remarkable performance in domains where previous safe hash algorithms failed because it is based entirely on different architectural notions for security than the safe hash algorithms (SHA-0 and SHA-1). With its adaptable architecture, high overall performance, and significant security margin, Keccak was an excellent choice. To get beyond the Memory Corruption Units (MCUs), new per word Decimal Matrix Code (DMC) was used. In order to find and fix more problems, this protection code used a decimal algorithm. According to the results of the implementation, the used technique offers better protection against big MCUs in memory.

Dilli, R., & Reddy, P. C. S. (2016) In terms of Throughput, Packet Delivery Fraction, and End-to-End Delay metrics, SHA3-512 outperforms SHA3-256 in the Zone Routing Protocol simulations. If these authors take the provided 64-bit CPU and compare the two methods' hash code calculation speeds, authors find that SHA3-512 was about 50% faster than SHA3-256. However, as the length of the hash value increases, more storage hardware is required

for SHA3-512.

Gowthaman, A., & Sumathi, M. (2015) these authors used Quartus-V9.0 to study the architecture of the current and proposed SHA-256 algorithm and then implements it in software. High performance, throughput, and efficiency are shown by the suggested / updated design. Scheduling computations inside the inner loop of the SHA256 algorithm was the basis of the hardware architecture. To make the SHA-256 function more secure, the suggested design made use of a Non-Linear Pseudo random generator. The 8-bit, 32-bit problematic hash result was generated by rearranging the pseudo code sequence generator. For each repetition, the delay was minimized using the Carry Select Adder.

Kaur, V., & Singh, A. (2015) the three pillars of security—confidentiality, authenticity, and integrity—are all provided by a suggested encryption scheme. For privacy, these authors used Advanced Encryption Standard (AES), and for authenticity and integrity checking, we use Rivest Shamir Adleman (RSA) and SHA-512, respectively. Significant security was provided by encrypting the AES key using RSA. Adding RSA to the hybrid scheme makes it more difficult for an adversary to break. Because it uses symmetric keys, AES was resistant to square attacks and provides fast processing. Concerning symmetric key algorithms like AES, key management was the biggest challenge. To address this issue, RSA was used. When both fast computing and security are paramount, this hybrid method shines.

Kotel, S., &Sbiaa, F. (2022) Elliptic Curve Cryptography (ECC) was regarded a very efficient option because to its enhanced performance in terms of computational power and battery resource needs; nonetheless, there have been other efforts to create a secure Cloud environment. ECC offered a safe and reliable methodology for creating and releasing encrypted apps to the cloud. Guarantee the safety of data stored in cloud datacenters, these authors presented a crypto-system. A hybrid approach is using a novel implementation of ECC functions in conjunction with One-Time Pad(OTP) Symmetrical Encryption Method and the SHA-3 algorithm is the primary contributed in these authors research.

Rana, K., et al. (2020) feel safe when it comes to data management and outsourcing, as well as the storage and protection of sensitive user information, cloud computing was indispensable. Limited security and hashing schemes have been implemented in several algorithms. For the sake of security, the Hyper Elliptic Curve (HECC) asymmetric public-key cryptography approach was used. With the aid of the user's signature at the storage end offers encryption and decryption. When used correctly, the SHA-2 hashing algorithm facilitates one-way compression with indexing. The suggested technique, which includes a cloud component, was delivered with efficient Software as a Service (SaaS) application.

Ravilla, D., &Putta, C. S. R. (2015) these authors working on two strategies to increase network security and avoid Denial of Service attacks: Hash based Message Authentication Code (HMAC)-SHA256 for data integrity and authentication, and a Trust-Based system. These authors rigorously matched the protocol's functionality to low-cost cryptographic primitives, including the hash function (HMAC-SHA256), resulting in a robust protocol capable of withstanding multiple network attacks.

Sundaram, B. B., et al. (2021) these authors proposed zone hierarchical routing protocol for Mobile Ad hoc Networks (MANETs) attempted to solve the problem of making such networks more secure using this method. The Zone Based Hierarchical Link State Routing Protocol (ZHLS) is a zone-based routing system that routes based on node and zone Intrusion Detection (IDs). Both Inter-zone Routing Protocol (IERP) and Intra-zone Routing protocol (IARP) immersed the same two routing systems. The proposed protocol immersed cryptographic security methods such as AES, SHA256, and Deffi-Hellman to ensure route safety inside and outside the zone, as well as in adjacent zones.

Natho, P.,et al(2024) to attempt to avoid hacking, passwords is frequently asked using hash algorithms to make them harder to decipher. These methodsutilized a variety of hash methods, including message digest (MD5) and secure hash algorithms. Their vulnerability derives from people using them to store passwords. A rainbow table is one possible strike. The Hashcat application can evaluate passwords created using a variety of hash approaches based on various hash algorithms. Regarding password storage, the SHA3 algorithm has shown to be safer than others

Table 1: Comparison of Secure Hashing Algorithms

Authors/Years	Algorithms Used	Purpose	Key Findings
Gowthaman& Sumathi (2015)	SHA-256 with Non-Linear PRNG	Enhanced SHA-256 design	Improved SHA-256 architecture using pseudo-random generators and optimized delay for higher performance.
Kaur & Singh (2015)	AES, RSA, SHA-512	Hybrid encryption for data security	RSA secures the AES key, enhancing both confidentiality and speed; suited for fast and secure computing.
Dilli & Reddy (2016)	SHA3-512, SHA3-256	Simulation comparison in ZRP	SHA3-512 outperforms SHA3-256 in throughput and packet delivery, but requires more storage.
Rana et al. (2020)	HECC, SHA-2	Cloud data protection	HECC with SHA-2 for efficient SaaS applications, enhancing encryption and signature-based authentication.
Kotel &Sbiaa (2022)	ECC, OTP, SHA-3	Cloud security	ECC combined with OTP and SHA-3 enhances secure cloud data storage and app encryption.
Natho, P., et al (2024)	MD5, SHA	Hardening password storage by salting and hashing	Emphasizes the need for stronger hash algorithms (like SHA3) in password storage due to vulnerabilities of MD5 and SHA1

2.1 Problems Identification

Data security and authentication in Wireless Sensor Networks (WSNs) are problematic due to high power consumption, large bandwidth requirements, and complex data processing. We are solving these issues by using hash-based intrusion detection algorithms (SHA2 and SHA3) to improve hardware and software performance as well as secure data transfers. AES is used to secure and authenticates the data. These systems provided solutions to critical security challenges such as authentication, integrity, and secrecy.

3. Materials and Methods

This research combines to enhance authentication and data security with secure hash-based intrusion detection in wireless sensor networks. The demands of the application will dictate whether sensor nodes should be placed in a grid or at random. The sensors deliver data

encrypted with SHA-2 or SHA-3, depending on the node's distance from the base station. SHA2 is utilized for fast and secure data transfers, whereas SHA3 is working in cryptographic applications. Security validation compares hash values to predicted hashes. In this research, we are aiming to all sending data will be secure. It warns the user of the danger of aggressive action in the event of a mismatch. Following that, it takes action to protect critical data by detecting the danger and assuring network security. The technology safeguards private data in response to intrusion detection during network monitoring. This technology safeguards data flow and network security by using cryptographic hashes, which detect any suspicious activity or changes to network security. Figure 1 shows the step by step process of data security and authentication process.

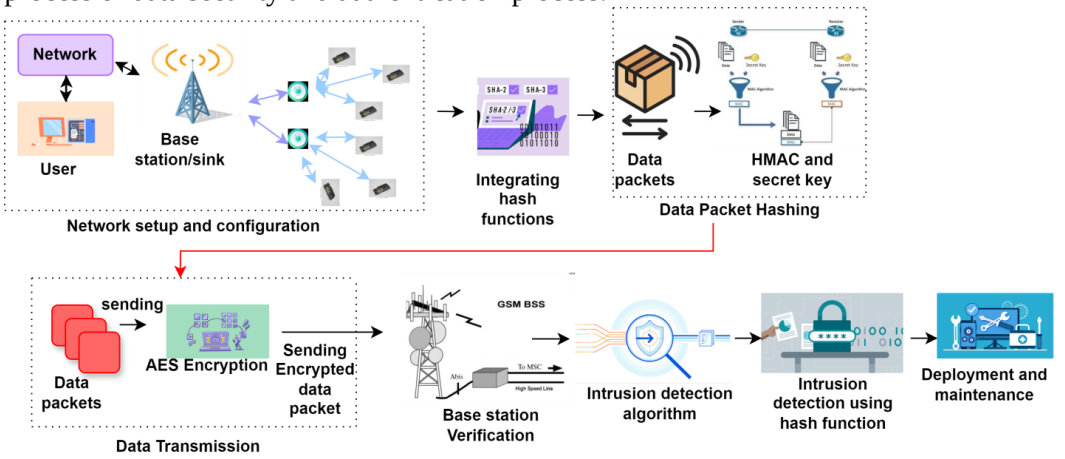

Figure 1: Overall architecture

3.1 Implementing Secure Hash based Intrusion Detection System

Security is one of the most problematic aspects of wireless networks, particularly those that are ad hoc or sensor networks. Because their architecture is less robust and more dynamic than that of wired networks, they are more vulnerable to attacks. According to Kumari, S., et al [24] WSNs may be seen as a subclass of ad hoc networks with restricted or no mobility.

- Sensor nodes are placed over a field to monitor local activities.
- Sensor nodes forward data to a sink or base station.
- Nodes are self-organizing in WSNs.

A sensor node consists of a memory unit, a CPU unit, a sensing unit, and a communication unit. It uses its sensory skills to navigate its environment. It stores data in the memory unit for a set amount of time and the processing unit facilitates data management. Finally, it delivers this information one hop at a time to the base station. In this case, the sensor node functions independently. According to Azeez, N. A.,[4] an Intrusion Detection System (IDS) monitors network traffic and detects nodes that aren't operating normally. While WSN research is still in its early stages, wired networks, also known as ad hoc networks, are well-studied. IDS are another item that may be installed on either the client or the server. We call this unit IDS agent. IDS agents monitor network behavior, detect intrusions, and respond to abnormal activity in three fundamental sequential steps.

A cryptographic hash function creates a fixed-size bit string from an arbitrary block of data

in such a way that any (accidental or intentional) changes to the input will almost certainly influence the hash output. The ideal cryptographic hash function is defined by four key properties.

- Creating a message with a specific hash is difficult; nevertheless, the hash value for every given message is calculated rapidly.
- Changing a message requires concurrently changing its hash.
- Two separate messages with the same hash would be undetectable.
- A Secure Hash-Based Intrusion Detection (SHID) system paired with a Wireless Sensor Network (WSN) ensures that data shared between users is legitimate and secure.

Given the current situation, modern data security is more important than ever. Hackers attack the user by using illegally acquired sensitive and personal data. As a result, we ensure that our data is never disclosed or is transferred securely (according to Bhavani, A., [15]). In this study, we combine WSN with SHID to increase data security and authentication. We initially construct and organize the network. Integrating the hash algorithms SHA2 and SHA3 provides for secure data transit. By generating cryptographic hash values for a variety of security applications—digital signatures, password storage, HMAC, SSL/TLS certificates, and block-chain transactions—SHA-2 and SHA-3 ensure data integrity, authenticity, and security. Hash based Message Authentication Code (HMAC) ensures that the message has not been modified and thereby verifies the identity of the message sender, assisting in the protection of communications, validation of data veracity, and prevention of tampering—all of which are critical components of modern data security systems. Following then, input is provided via network packets or sensor data. They provide important information about network or system activities, such as IP addresses, port numbers, payload size, and time stamps. These are called data packets. Data is represented as D = $(d_1, d_2, d_3, \dots \dots d_n)$.

A secure hash function (SHA-256 or SHA-3) is applied to the extracted feature vectors. A cryptographic hash function maps data to a fixed-size string of characters.

$$h_i = H\ (d_i) \text{ -------- } (1)$$

In equation (1) H is a Hash function, $h_i$is the hash value corresponding to feature vectord_i.

Hashing data packets are encrypted using AES algorithm then, send these data packets to base station.

A baseline or reference hash is created using normal or expected data flows. This baseline can be obtained by hashing known good behaviors or traffic patterns in equation (2).

$$h_b = H\ (d_b) \text{ ------- } (2)$$

Where $d_b$represents the normal feature set and h_b is the hash corresponding to normal network activity.

Each incoming data flow is compared with the baseline hash. A significant deviation between the hashes can indicate an intrusion attempt. The difference can be quantified using a similarity or distance measure (in equation 3).

$$\mathrm{X}(h_i, h_b) = \sum_{k=1}^{l} |\, h_i[k] - h_b[k]| \text{ --------- } (3)$$

Where, X is the difference between hash of the current data and the baseline. $h_i[k], h_b[k]$ are the responsive hash values and l is length of the hash.

When an intrusion is detected, the system may trigger alarms or take defensive actions such as isolating the compromised node or logging the intrusion for further analysis.

Table 2: comparison of SHA2 and SHA3 hashing algorithms

Feature	SHA-2	SHA-3
Hash Lengths Available	224, 256, 384, 512 bits	224, 256, 384, 512 bits
Internal Structure	Merkle-Damgård construction	Sponge construction
Security Level	Secure, but shares design principles with older algorithms (like SHA-1)	Newer design believed to be more resistant to certain attacks
Speed and Performance	Generally faster on most hardware	Typically slower than SHA-2
Use Cases	Digital signatures, SSL/TLS, crypto-currency	Digital signatures, password hashing, cryptographic applications
Collision Resistance	Strong (no known practical collisions)	Strong (designed to prevent collisions)
Flexibility	Fixed output length per algorithm variant	Variable output lengths and extendable-output functions (XOFs)
Post-Quantum Security	Not specifically designed for post-quantum resistance	Considered more resilient against quantum attacks due to its unique design
Adoption	Widely adopted across various security protocols	Increasing adoption, particularly in new applications and standards

3. 2 Steps for Secure Hash-Based Intrusion Detection

3.2.1. Structure the Network Setup and Configuration

The needs of the application will determine whether sensor nodes are placed randomly or in an ordered grid. Set dependent on node-to-base station distance. Long-range WSNs may need a range of up to 200 meters, whilst short-range WSNs may find a range of 10-50 meters enough. Base stations are often utilized as central servers or nodes that collect and process data from sensor nodes. One may compare it to external data analysis systems. It handles data security, aggregation, and routing. It should also support organized clusters, including Cluster Heads (CH), sensor nodes, and hash-based intrusion detection verification. Before passing it to the base station, the CH collects data from individual nodes. This reduces energy consumption and network traffic. TDMA (Time Division Multiple Access) is recommended to reduce collisions and energy usage in synchronized networks, with each node assigned a distinct time slot for communication. Control the range of each node by adjusting its transmission strength. Ideal range depends on network density and deployment location. A greater range consumes more power but reduces the need for intermediate relays, while a shorter range conserves energy but requires more hops. Use homomorphic encryption and other cryptographic approaches to allow aggregation while maintaining data confidentiality. Using energy-aware routing algorithms choose routes that use less energy while ensuring reliable data transport.

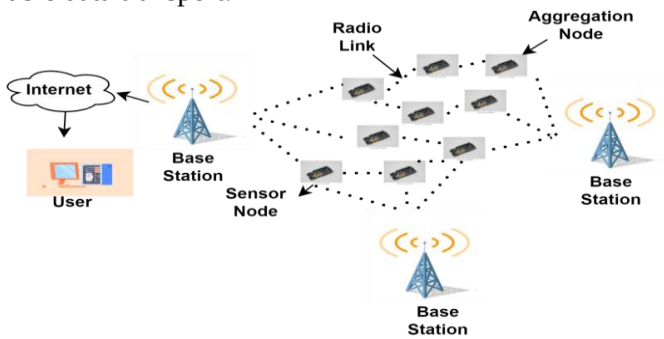

Figure 2: Network setup and configuration

2. Integration of Cryptographic Hash Functions and Data Packet Hashing

Typically, this approach employs hash functions to provide authentication, confidentiality, and data integrity during base station-to-sensor node communication. Check to see whether the firmware contains the cryptography library with the selected hash function. Choose a secure hash algorithm, such as SHA-256 or SHA-3, based on your security requirements and performance. Collect data from the sensors. This might be another environmental metric, such as temperature or humidity. The sensor node's firmware or software should have libraries or algorithms for the specified hash functions. This permits easy hash operations between sensor nodes. Create a data packet as if it were a little box containing source node ID, contextual information, sensor node metadata, timestamp. Collect all of the essential data: Interpret the sensor data. Obtain the timestamp and node ID. The hash function will take one string or byte array as input, which will include all of the data packet's contents. Calculating the hash of the obtained data ensures that any data changes are evident. Combine the hash function with a secret key to provide even greater security using HMAC. This ensures that the data has not been changed, and thereby authenticates the data sender. Send hash, HMAC, and sensor data to the base station. When the base station receives data, it must check the HMAC and hash. Calculate the hash at the base station and compare it with the received hash. Check the HMAC to ensure that the data source is legitimate.

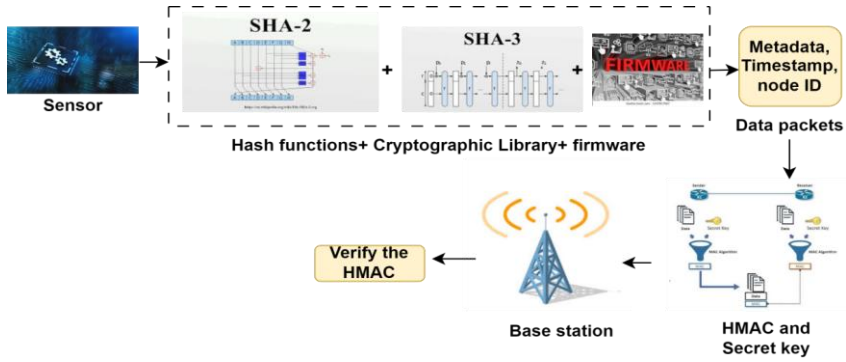

Figure 3:Integration of Cryptographic Hash Functions and Data packet hashing process

3. Secure Data Transmission

Before transmitting every data packet, calculate its hash using a cryptographic hashing tool. Its hash serves as the data's fingerprint, ensuring that any changes to the content may be detected. Securing the data during transmission using AES encryption technologies ensures that illegitimate users cannot access it. AES was also used in large data encryption due to its security and speed. Get the base station to which the encrypted data packet was delivered. Although the information is encrypted, this packet contains both the hash and the contents.

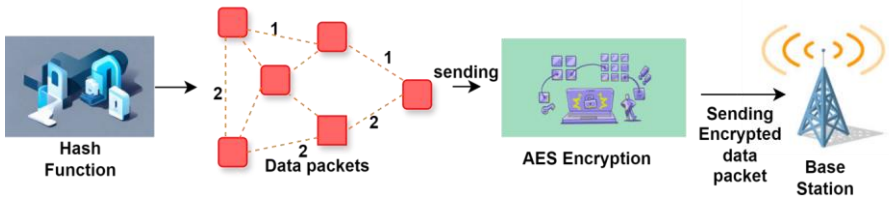

Figure 4: Data Transmission using AES

4. Base Station Verification

Data packets encrypted from sensor nodes make their way to the base station. First, decode the incoming data packet to get the original content; secondly, hash it using the same encryption key used for encryption. Decryption then isolates the actual data from the hash. Paragraphing the encrypted data may assist get the content and hash value. Recalculate the hash using the same hash algorithm and the original data. Compare the hash produced again to the hash obtained from the decrypted packet. If the hashes match, the data is confirmed to be valid and complete. If they do not match, it is possible that the data was changed during transmission.

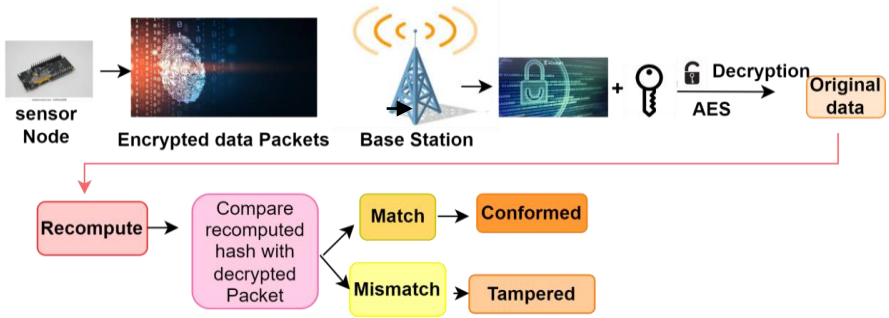

Figure 5: Base Station verification

5. Intrusion Detection Algorithms

Intrusion detection is to monitor network traffic and identify any odd behavior that suggests malicious activity. Using hash functions, establish a baseline for typical behavior. Examine and record the hash values of data packets over time while the network functions normally. Create and store hash values in a repository; they will be used for comparison. Establish realistic hash value deviation thresholds based on a research of typical conduct. These features include tolerance and detection sensitivity. The hash values of incoming data packets are computed in real time by hashing them as they arrive at the base station. By comparing the incoming hash values to the traditional hash repository, one may determine if they are within the allowable range of deviation. Inform network management if an incoming hash deviates considerably from the baseline by notifying or alerting systems to potential security issues. Sort the hash as unusual or suspicious.

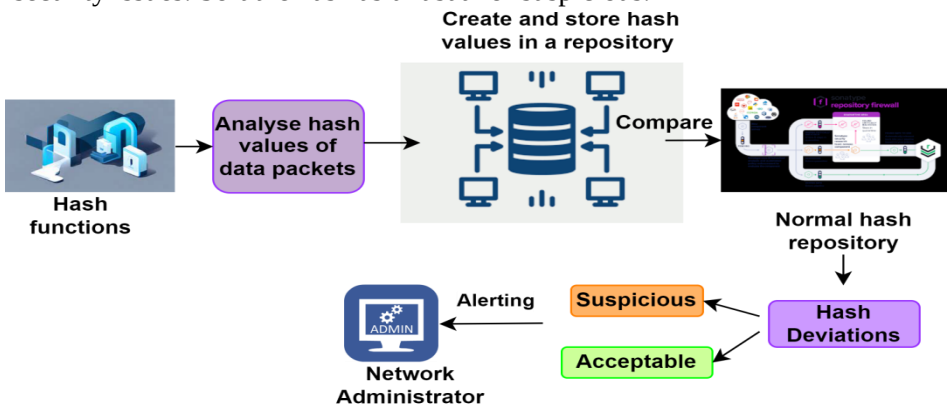

Figure 6: Intrusion Detection using hash functions

6. Deployment and Maintenance

Base station and sensor node deployments should be planned and executed with coverage and surrounding situations in mind. Before proceeding with the deployment, ensure that the network is performing optimally.

Maintenance

Regular maintenance and monitoring help you keep your network error-free and secure. Upgrading equipment and software as needed may assist in addressing any security problems; also, providing staff data backups for use during an emergency. This technology ensures the integrity and security of data over time by using a robust and secure wireless sensor network.

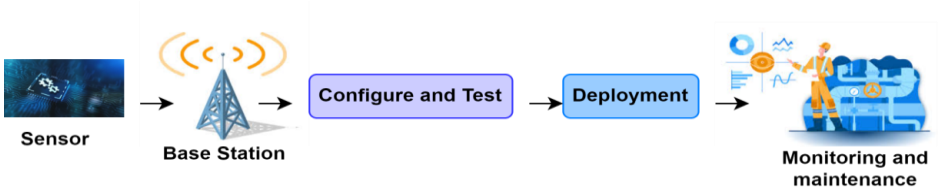

Figure 7: Deployment and maintenance

Algorithm 1: Secure Hash-Based Intrusion Detection

Input:

- Data packet D_i to be transmitted.
- Secret key K_i for each sensor node.
- Hash function $H()$.

Steps:

- Calculate the hash value $H(D_i)$ for the data packet D_i using the hash function H().
$$H(D_i) = H(data)$$
- Including the hash value $H(D_i)$ in the data packet (D_i) necessitates adjustments.
- Ensure privacy by encrypting the accompanying data packet using the secret key K_i .
$$D_i' = E(K_i, D_i \,||\, H(D_i))$$
- Transmit the encrypted data packet D_i' throughout the wireless sensor network.
- Get the encrypted data packet D_i at the destination node.
- Decrypt the packet using the corresponding secret key K_i.
$$(D_i \,||\, H(D_i)) = D_i' = D(E - 1(K_i, D_i'))$$
- Recomputed the hash value $H(D_i)$ for the received data D_i.
- Compare the hash value acquired in the packet to the recomputed one.
$$H(D_i)_{received} =^{?} H(D_i)_{computed}$$
- If the hash values match, the data integrity stays intact and no intrusion is discovered.
- If the hash values vary, an intrusion or manipulation is detected, triggering an alert.

Output:

Finding any modification or intrusion in the sent data stream.

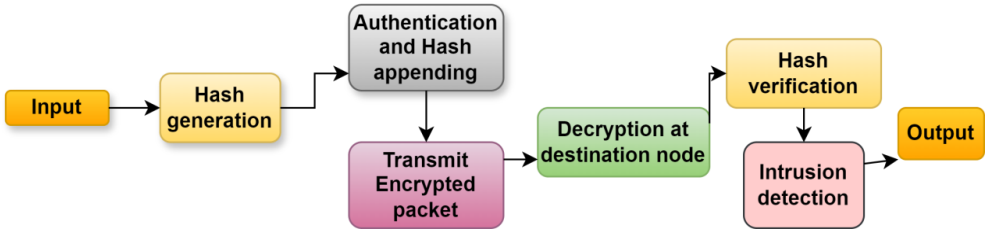

Figure 8: Flow diagram for Hash based Intrusion detection System

The Secure Hash-based Intrusion Detection system guarantees data integrity over packet flow in a wireless sensor network. A secure hash function computes the hash value of a generated data packet. This hash serves as the data packet's unique identifier. The generated hash value concludes the data packet. The whole packet data and hash is encrypted with a secret key that is unique to each sensor node, ensuring that the data remains private during journey. The encrypted data packet was sent to the target node over the wireless sensor network. The encrypted packet is retrieved and decoded using the matching secret key, revealing both the hash and the contents. The destination node calculates the hash value from the data it receives. It then compares the newly derived hash to the one in the packet. If the two hash values match, the data is unmodified and not tampered with. If there is a disparity, it indicates suspected infiltration or manipulation and triggers an alert.

4. Results and Discussions

In this study, we are using NS2 to implement Secure Hash-based Intrusion Detection (SHID) system for enhancing data security and authentication. The proposed SHID performs well compared to other existing algorithms. While the proposed SHID system have a shorter time delay and energy usage than the existing hash algorithms, the throughput and Packet Delivery Ratio (PDR) are high compared to other existing hash algorithms.

4.1 Performance evaluation

4.1.1 Throughput

Throughput computation refers to the process of assessing how efficiently data is transmitted via a network or system at a given moment. The information flow of a data transmission is often shown in bits per second (bps), kilobits per second (Kbps), Megabits per second (Mbps).

$$\text{Throughput} = \frac{N}{T \times S} \quad \text{---------- (4)}$$

N= Number of packet size, T= Time duration, S= Successful average Packet size

Table 3: Throughput comparison table

	Throughput levels			
Packet Size	**SHA1** [Natho, P., et al (2024)]	**SHA2** [Kadhim, R. A., (2015)]	**SHA3** [Asha, T. R., (2015)]	**Secure Hash based Intrusion detection (SHA2 with SHA3)**
50	0.312	0.345	0.365	0.412
100	0.489	0.523	0.567	0.799
150	0.796	0.797	0.821	1.234
200	1.102	1.123	1.167	1.689
250	1.213	1.234	1.267	2.112

Table 3 compares with different packet size the throughput value of many hash algorithms (SHA1, SHA2, SHA3 and Secure Hash based Intrusion Detection System). The SHIDS has the highest throughput, starting at 0.412 units and increasing to 2.112 units at the maximum packet size. With the SHIDS beating the others, this suggests that bigger packet sizes usually enhance the efficiency of data processing across different methods, hence managing data security while preserving high throughput.

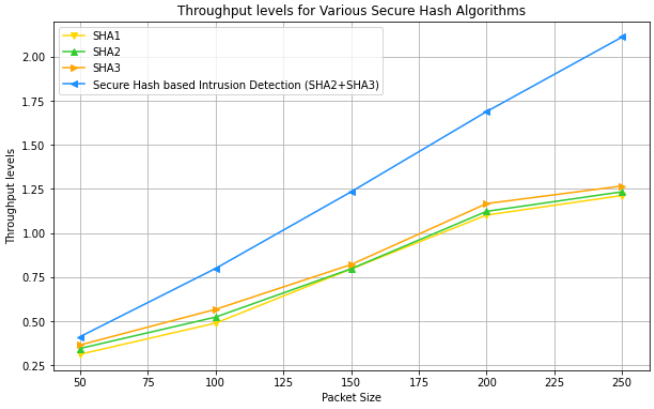

Figure 9: Throughput value comparison of various hashing algorithms

Figure 9 shows, across multiple packet sizes ranging from 50 to 250 bytes, the throughput levels of several secure hash algorithms—SHA1, SHA2, SHA3, and a Secure Hash-based Intrusion Detection System (SHIDS) are combining SHA2 and SHA3. Each packet size produces a higher throughput figure than any other existing approach. The y-axis of this chart shows throughput data, while the x-axis depicts packet sixe.

4.1.2 Energy Consumption

It refers to the evaluation of a system's or device's energy consumption for task performance. Energy performance is a critical element in networking and computing, particularly in resource-constrained environments such as WSNs and IoT devices, which rely on battery-powered devices.

$$\text{Energy}= \frac{NS}{E} \times 100 \quad \text{--------- (5)}$$

NS = Number of sensor nodes, E = Energy consumption for sending packets at a times

Table 4: Energy comparison table

	Energy level in joules			
Number of Nodes	**SHA1** [Natho, P., et al (2024)]	**SHA2** [Kadhim, R. A., (2015)]	**SHA3** [Asha, T. R., (2015)]	**Secure Hash based Intrusion detection (SHA2 with SHA3)**
10	62	55	51	45
20	110	102	100	92
40	220	204	196	167
60	353	342	328	295
80	456	442	406	381
100	611	560	518	497

Table 4 shows energy comparison value of various hashing algorithms (SHA1, SHA2, SHA3 and Secure Hash based Intrusion detection system) with various packet size. The SHIDS takes the low energy beginning at 45 with 10 packet sizes and surging to 497 units at the largest packet size. This indicates that larger packet sizes generally improve the efficiency of data processing across these algorithms, with the SHIDS outperforming the others, highlighting its effectiveness in managing data security while maintaining with low energy consumption.

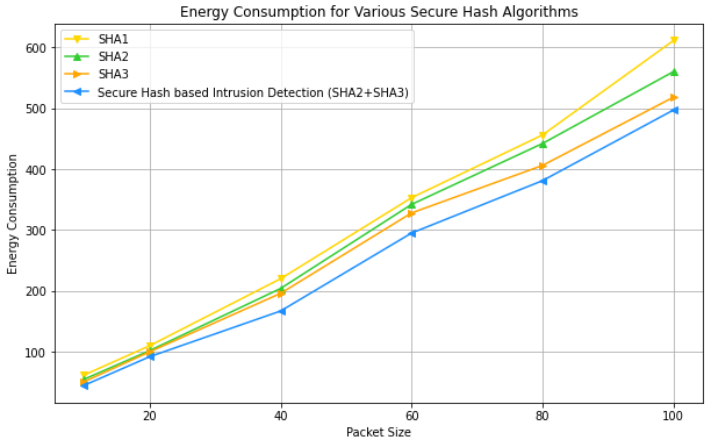

Figure 10: Energy consumption comparison of various hashing algorithms

Figure 10 illustrates Energy consumption of various secure hash algorithms—SHA1, SHA2, SHA3, and a Secure Hash-based Intrusion Detection System (SHIDS) that combines SHA2 and SHA3—across different packet sizes ranging from 10 to 100 bytes. Energy consumption of proposed method is low compared to other existing algorithms. In this chart, the x-axis shows packet sixe and the y-axis shows Energy consumption.

4.1.3 Time Delay

Time delay seems to be calculated in a sensor network using sensor node count, energy utilization, and forwarding time during data transmission.

$$\text{Time Delay} = \frac{NS}{E \times T} \text{ ----- (6)}$$

NS= Number of sensor nodes, E=Energy consumption for sending packets at a times, T=Forwarding time in ms

Table 5: End to End delay comparison

	End to End Time Delay (ms)			
Number of Nodes	**SHA1** [Natho, P., et al (2024)]	**SHA2** [Kadhim, R. A., (2015)]	**SHA3** [Asha, T. R., (2015)]	**Secure Hash based Intrusion detection (SHA2 with SHA3)**
10	0.050	0.047	0.034	0.010
20	0.157	0.134	0.123	0.110
40	0.213	0.201	0.194	0.164
60	0.314	0.298	0.275	0.241
80	0.428	0.412	0.398	0.376
100	0.546	0.521	0.502	0.481

Table 5 shows time delay comparison of various hashing algorithms (SHA1, SHA2, SHA3 and Secure Hash based Intrusion detection system) with various packet size. The SHIDS demonstrates short time delay, beginning at 0.010ms with packet size 10 and surging to 0.481ms with 100 packet sizes. This indicates that larger packet sizes generally improve the efficiency of data processing across these algorithms, with the SHIDS outperforming the others, highlighting its effectiveness in managing data security while maintaining low time delay.

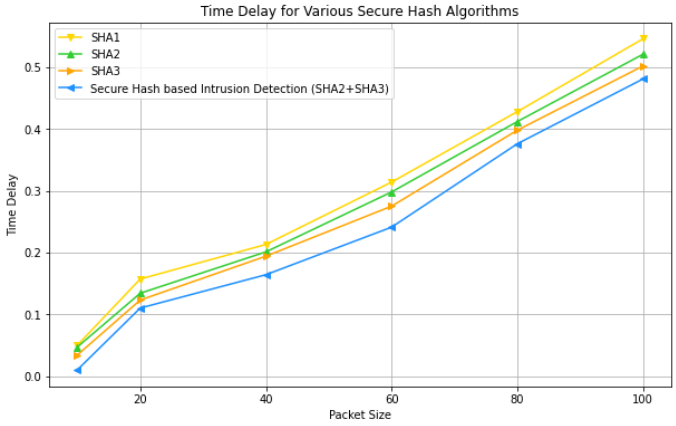

Figure 11: Time delay comparison of various hashing algorithms

Figure 11 illustrates the time delay of various secure hash algorithms—SHA1, SHA2, SHA3, and a Secure Hash-based Intrusion Detection System (SHIDS) that combines SHA2 and SHA3—across different packet sizes ranging from 10 to 100 bytes. The proposed method takes low time compared to other existing algorithms. In this chart, the x-axis shows packet sixe and the y-axis shows delay time values.

4.1.4 PDR

In networking, the Packet Delivery Ratio (PDR) is a performance metric of data transit reliability. It displays the fraction of successfully received packets to the total packet count. Because more packets are efficiently delivered, a higher PDR indicates improved network performance.

$$PDR = \frac{R}{T} * 100 \quad \text{------- (7)}$$

R = Total number of packets receives, T = Total packet

Table 6: Packet delivery ratio

	Packet Delivery Ratio			
Number of packets	**SHA1** [Natho, P., et al (2024)]	**SHA2** [Kadhim, R. A., (2015)]	**SHA3** [Asha, T. R., (2015)]	**Secure Hash based Intrusion detection (SHA2 with SHA3)**
50	97.6	98.03	98.48	98.98
100	98.10	98.34	98.75	99.34
150	98.30	98.63	98.99	99.67
200	99.06	99.15	99.23	99.79
250	99.20	99.43	99.69	99.95

Table 6 examines the packet delivery percentage of numerous hash algorithms with varying packet sizes (SHA1, SHA2, SHA3, and Secure Hash-based Intrusion Detection System). The SHIDS has the highest PDR values, starting at 98.98 with a packet size of 50 units and increasing to 99.95 units at the maximum packet size. With SHIDS outperforming the others, it seems that larger packet sizes often improve the efficiency of data processing across various approaches, implying that regulating data security while maintaining high PDR values is definitely effective.

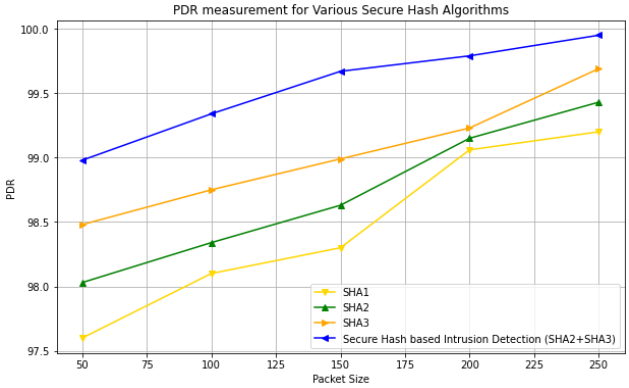

Figure 12: PDR value comparison of various hashing algorithms

Figure 12 illustrates the PDR value comparison of various secure hash algorithms SHA1, SHA2, SHA3, and a Secure Hash-based Intrusion Detection system (SHID) that combines SHA2 and SHA3 across different packet sizes ranging from 50 to 250 bytes. In all packet sizes, the PDR value exceeds that of other known techniques. In this chart, the x-axis shows packet sixe and the y-axis shows PDR values.

5. Conclusion

In conclusion, a Secure Hash-based Intrusion Detection System (SHID) concerns about the integrity of kept data and the security of Wireless Sensor Networks. By use of secure hash functions to identify and eliminate threads and illegal access, the suggested intrusion detection system (IDS) guarantees the security and authenticity of given data. This improves WSN security design and provides a consistent method for managing odd behavior. WSN installations in military applications, environmental monitoring, and healthcare will be more trustworthy if network security is approached proactively against different threats. With this method, we ensure data transfers over wireless networks remain secure. The findings demonstrate the performance of the hashing approach far better than the present hash techniques in use is the secure hash-based intrusion detection system. With 250 packets size from SHID had PDR ratings of 99.95% and the throughput value is 2.112%. Conventional approaches need more time and work than this one. Among the interesting characteristics of packet size 100 are a time delay of 0.481 and an energy value of 497. In future these hash functions will apply to assist wireless sensor networks with secure data transmission and authentication of data.

References

1. Ali, R., Pal, A. K., Kumari, S., Karuppiah, M., & Conti, M. (2018). A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. *Future Generation Computer Systems*, *84*, 200-215.
2. Al-Mashhadi, H. M., Abdul-Wahab, H. B., & Hassan, R. F. (2014, June). Secure and time efficient hash-based message authentication algorithm for wireless sensor networks. In *2014 Global Summit on Computer & Information Technology (GSCIT)* (pp. 1-7). IEEE.
3. Asha, T. R., &Hamsaveni, N. (2015). Implementation of Sha-3 for Security and Error Detection and Correction Mechanism to Enhance Memory Reliabilty. *International Journal of Engineering Research Technology (IJERT), ISSN*, 2278-0181.

4. Azeez, N. A., Ayemobola, T. J., Misra, S., Maskeliūnas, R., &Damaševičius, R. (2019). Network intrusion detection with a hashing based apriori algorithm using Hadoop MapReduce. *Computers*, *8*(4), 86.
5. Bahache, A. N., Chikouche, N., &Mezrag, F. (2022). Authentication schemes for healthcare applications using wireless medical sensor networks: A survey. *SN Computer Science*, *3*(5), 382.
6. Bhavani, A., & Nithya, V. (2023). Cryptographic algorithm for enhancing data security in wireless IoT sensor networks. *Intelligent Automation & Soft Computing*, *36*(2), 1381-1393.
7. Butun, Ismail, (2013) "Prevention and Detection of Intrusions in Wireless Sensor Networks" . Graduate Theses and Dissertations. http://scholarcommons.usf.edu/etd/4449
8. Choi, Y., Lee, D., Kim, J., Jung, J., Nam, J., & Won, D. (2014). Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors*, *14*(6), 10081-10106.
9. Darbandeh, F. G., &Safkhani, M. (2023). SAPWSN: A secure authentication protocol for wireless sensor networks. *Computer Networks*, *220*, 109469.
10. Deng, J., Han, R., & Mishra, S. (2006, April). Secure code distribution in dynamically programmable wireless sensor networks. In *Proceedings of the 5th international conference on Information processing in sensor networks* (pp. 292-300).
11. Dilli, R., & Reddy, P. C. S. (2016, October). Implementation of security features in MANETs using SHA-3 standard algorithm. In *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)* (pp. 455-458). IEEE.
12. Ferng, H. W., & Khoa, N. M. (2017). On security of wireless sensor networks: a data authentication protocol using digital signature. *Wireless Networks*, *23*, 1113-1131.
13. Gope, P., & Hwang, T. (2016). A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. *IEEE Transactions on industrial electronics*, *63*(11), 7124-7132.
14. Gowthaman, A., & Sumathi, M. (2015). Performance study of enhanced SHA-256 algorithm. *International Journal of Applied Engineering Research*, *10*(4), 10921-10932.
15. Hodowu, D. K. M., Korda, D. R., &Ansong, E. D. (2020). An enhancement of data security in cloud computing with an implementation of a two-level cryptographic technique, using AES and ECC algorithm. *Int. J. Eng. Res. Technol*, *9*, 639-650.
16. Hussain, M., Mehmood, A., Khan, S., Khan, M. A., & Iqbal, Z. (2019). Authentication techniques and methodologies used in wireless body area networks. *Journal of Systems Architecture*, *101*, 101655.
17. Ibrahim, R. K., Kadhim, R. A. J., &Alkhalid, A. S. H. (2015, September). Incorporating SHA-2 256 with OFB to realize a novel encryption method. In *2015 World Symposium on Computer Networks and Information Security (WSCNIS)* (pp. 1-6). IEEE.
18. Kadhim, R. A., Ibrahim, R. K., &Alkhalid, A. S. (2015). Implementation of Secure Hash Algorithm Sha-2 256 by using Labview. In *International Conference on Image Processing, Production and Computer Science* (pp. 112-119).
19. Kamalesh, S., & Kumar, P. G. (2017). Fuzzy based secure intrusion detection system for authentication in wireless sensor networks. *Journal of Computational and Theoretical Nanoscience*, *14*(5), 2465-2472.
20. Kaur, V., & Singh, A. (2015). An encryption scheme based on AES and SHA-512. *International Journal of Applied Engineering Research*, *10*(10), 25207-25218.
21. Khalid, B., Qureshi, K. N., Ghafoor, K. Z., & Jeon, G. (2023). An improved biometric based user authentication and key agreement scheme for intelligent sensor based wireless communication. *Microprocessors and Microsystems*, *96*, 104722.
22. Khan, M. K., &Alghathbar, K. (2010). Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. *Sensors*, *10*(3), 2450-2459.

23. KOTEL, S., & SBIAA, F. (2022). A Data Security Algorithm for the Cloud Computing based on Elliptic Curve Functions and Sha3 Signature. *International Journal of Advanced Computer Science and Applications*, *13*(3).
24. Kumari, S., Khan, M. K., &Atiquzzaman, M. (2015). User authentication schemes for wireless sensor networks: A review. *Ad Hoc Networks*, *27*, 159-194.
25. Natho, P., Somsuphaprungyos, S., Boonmee, S., &Boonying, S. (2024) Comparative study of password storing using hash function with MD5, SHA1, SHA2, and SHA3 algorithm. *Int J Reconfigurable & Embedded Syst ISSN*, *2089*(4864), 4864.
26. Njuki, S., ZHANG, J., TOO, E., & DADYE, H. B. (2019). Enhancing user data and vm security using the efficient hybrid of encrypting techniques. *Journal of Theoretical and Applied Information Technology*, *97*(15).
27. Patel, S. V., Pandey, K., & Rathod, V. R. (2008, December). Decentralised clustered and hash based intrusion detection system for wireless sensor networks. In *2008 Fourth International Conference on Wireless Communication and Sensor Networks* (pp. 27-30). IEEE.
28. Raj, A. B., Ramesh, M. V., Kulkarni, R. V., &Hemalatha, T. (2012, June). Security enhancement in wireless sensor networks using machine learning. In *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems* (pp. 1264-1269). IEEE.
29. Rajeswari, S., Arunmozhi, S. A., &Venkataramani, Y. (2022). Wireless Sensor-Based Hashing Technique for Secure Patient Record Transferring In Biometrics System. *NeuroQuantology*, *20*(11), 485.
30. Rana, K., Yadav, H., & Agrawal, C. (2020, March). Mutual Authentication and Location Privacy using HECC and SHA 2 in Mobile Cloud Computing Environment. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 362-369). IEEE.
31. Ravilla, D., &Putta, C. S. R. (2015, January). Implementation of HMAC-SHA256 algorithm for hybrid routing protocols in MANETs. In *2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV)* (pp. 154-159). IEEE.
32. Roy, P. K., & Bhattacharya, A. (2022). SDIWSN: A software-defined networking-based authentication protocol for real-time data transfer in industrial wireless sensor networks. *IEEE Transactions on Network and Service Management*, *19*(3), 3465-3477.
33. Saleh, Y. N., Chibelushi, C. C., Abdel-Hamid, A. A., & Soliman, A. H. (2020). Privacy preservation for wireless sensor networks in healthcare: State of the art, and open research challenges. *arXiv preprint arXiv:2012.12958*.
34. Selvam, P. M., & Sujatha, S. S. (2020). A study on integrity and authentication using rsa and sha-3 algorithms for secured data communication. *International Journal of Engineering, Science and Mathematics*, *9*(8), 1-18.
35. Singh, D., Kumar, B., Singh, S., & Chand, S. (2020). Evaluating authentication schemes for real-time data in wireless sensor network. *Wireless Personal Communications*, *114*(1), 629-655.
36. Sklavos, N., &Koufopavlou, O. (2003, May). On the hardware implementations of the SHA-2 (256, 384, 512) hash functions. In *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03.* (Vol. 5, pp. V-V). IEEE.
37. Soni, D., & Mishra, N. (2017). Multilevel Authentication based Data Security and Verification over Cloud Computing Environment. *International Journal of Education and Management Engineering*, *7*(5), 56.
38. Sravani, M. M., &Durai, S. A. (2022). On efficiency enhancement of SHA-3 for FPGA-based multimodal biometric authentication. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *30*(4), 488-501.

39. Sundaram, B. B., Mishra, M. K., Thirumoorthy, D., Rastogi, U., &Pattanaik, B. (2021, July). ZHLS Security Enhancement by integrating SHA256, AES, DH in MANETS. In *Journal of Physics: Conference Series* (Vol. 1964, No. 4, p. 042003). IOP Publishing.
40. Sureshkumar, C., & Sabena, S. (2020). Fuzzy-based secure authentication and clustering algorithm for improving the energy efficiency in wireless sensor networks. *Wireless Personal Communications*, *112*(3), 1517-1536.
41. Tseng, H. R., Jan, R. H., & Yang, W. (2007, November). An improved dynamic user authentication scheme for wireless sensor networks. In *IEEE GLOBECOM 2007-IEEE global telecommunications conference* (pp. 986-990). IEEE.
42. Velmurugan, T., & Karthiga, S. (2020). Security based approach of SHA 384 and SHA 512 algorithms in cloud environment. *J Comput Sci*, *16*(10), 1439-1450.
43. Wang, D., Li, W., & Wang, P. (2018). Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks. *IEEE Transactions on Industrial Informatics*, *14*(9), 4081-4092.
44. Yeh, H. L., Chen, T. H., Liu, P. C., Kim, T. H., & Wei, H. W. (2011). A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors*, *11*, 4767-4779.
45. Yu, H., & Wang, L. (2019). A security-enhanced mutual authentication scheme with privacy protected in wireless sensor networks. *Cluster Computing*, *22*(Suppl 3), 7389-7399.
46. Nagarajan, M. "A New Approach to Improve Life Time Using Energy Based Routing in Wireless Sensor Network." International Journal of Science and Research (IJSR).
47. Nagarajan, M., and S. Karthikeyan. "A new approach to increase the life time and efficiency of wireless sensor network." International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012). IEEE, 2012.
48. Ezhilarasi, M., &Krishnaveni, V. (2019). An evolutionary multipath energy-efficient routing protocol (EMEER) for network lifetime enhancement in wireless sensor networks. Soft Computing, 23(18), 8367-8377.