A Mobile Multimedia Data Collection Scheme for Secured Wireless Multimedia Sensor Networks

Muhammad Usman, *Member, IEEE,* Mian Ahmad Jan*, *Member, IEEE,*
Xiangjian He*, *Senior Member, IEEE* and Jinjun Chen*, *Senior Member, IEEE*

Abstract—Wireless Multimedia Sensor Networks (WMSNs) produce an enormous amount of big multimedia data. Due to large size, Multimedia Sensor Nodes (MSNs) cannot store the generated multimedia data for a long time. In this scenario, mobile sinks can be utilized for data collection. However, due to vulnerable nature of wireless networks, there is a need for an efficient security scheme to authenticate both the MSNs and mobile sinks. In this paper, we propose a scheme to protect the underlying WMSN during mobile multimedia data collection. The proposed scheme is a two-layer scheme. At the first layer, the MSNs are distributed into small clusters, where each cluster is represented by a single Cluster Head (CH). At the second layer, the CHs verify the identities of the mobile sinks, before forwarding the multimedia data. Authentication at both these layers ensures a secure data exchange. We evaluate the performance of the proposed scheme through extensive simulation results. The simulation results prove that the proposed scheme performs better as compared to existing state-of-the-art approaches in terms of resilience and handshake duration. The proposed scheme is also analyzed in terms of authentication rate, data freshness, and packet delivery ratio, and has shown a better performance.

Index Terms—WMSNs, multimedia, MSNs, security, clusters, authentication.

1 Introduction

Traditional concept of the Internet of Things (IoT) consists of small sensing devices that are battery-operated and unable to perform long-range wireless communication [1]. The sensing devices usually form a small Wireless Sensor Network (WSN), and forward sensed data to a nearby Base Station (BS). In these networks, computationally-complex tasks are always performed at the BS. In recent years, evolutionary steps have been taken in the field of WSNs where audio and visual sensors are integrated into simple sensor nodes and have transformed them into Multimedia Sensor Nodes (MSNs). These nodes are able to store, correlate and fuse both multimedia and non-multimedia data [2]. The MSNs are gaining the attention of researchers across the globe due to their wide range of applications in daily lives, such as e-health, transport management system, and security and surveillance. For example, in smart cities, smart sensing cameras installed at major junctions, roads and highways can capture current scenarios of vehicular traffic, and forward the captured multimedia data to either local or remote cloud servers for further processing and analysis. Earlier generations of MSNs consist of a limited battery power and a low-power transceiver to transmit and receive multimedia data. However, latest generations, such as wireless and camera motes, have powerful hardware and are able to deal with high-resolution multimedia data. A comparison between the earlier and latest generations of MSNs was summarized in [3]. On a large scale, multiple smart sensing cameras operate simultaneously. As a result, huge volumes of multimedia data are generated, that need to be managed and processed efficiently in real-time.

Just like miniature sensors, the MSNs also build a network known as a Wireless Multimedia Sensor Network (WMSN). For an efficient data sharing and transmission, the WSNs/WMSNs are usually distributed into small clusters. Each cluster is represented by a nominated node, known as a Cluster Head (CH). The nominated CHs take responsibility for collecting the sensed/captured data from the member nodes and forwarding to the BS. Depending upon the underlying application, the captured data may contain sensitive information. Furthermore, there is usually no human intervention when the WMSNs are deployed in an open environment. Such deployments expose these networks to various network vulnerabilities and threats, such as node replication, Sybil attack, Denial-of-Service (DoS), and Distributed DoS (DDoS) attacks [4]. These attacks are usually launched by malicious nodes. Therefore, there is a need to protect these networks from vulnerabilities and threats through robust and highly efficient node authentication schemes.

For the MWSNs operating in open environments, a mobile sink can play an important role to collect and forward data to the local/cloud servers. The mobile sinks can be mobile sensors, smartphones or personal digital assistants, and are able to communicate with the nearby sensor nodes [5]. Therefore, the mobile sinks are an essential component in large-scale WMSNs. Sometimes, it is possible that the link between cloud/local servers and BS may be partially or completely damaged. As a result, the BS is able to control the network but unable to upload the collected data to the servers. On the other hand, the CHs continuously collect and forward the multimedia data to the BS. This data need to be offloaded by the BS due to the fact that

*A * indicates the corresponding author.*
Muhammad Usman and Jinjun Chen are with the Department of Computer Science and Software Engineering, Swinburne University of Technology, Australia. (E-mail: musman@swin.edu.au, jinjun.chen@gmail.com)
Xiangjian He is with the Global Big Data Technologies Center (GBDTC), School of Electrical and Data Engineering, University of Technology Sydney, Australia. (E-mail: xiangjian.he@uts.edu.au)
Mian Ahmad Jan is with the Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan (E-mail: mianjan@awkum.edu.pk)

the BS might not have enough storage space to store the forwarded real-time multimedia data for a long time. One solution to this problem is the use of mobile sinks. In this case, the CHs will forward the collected data to the mobile sinks, instead of forwarding to the BS. After receiving the data from CHs, the mobile sinks will forward the data to the local/cloud servers, as soon as they get access to the Internet. However, the use of mobile sinks to collect the data from the WMSNs introduces a new security challenge in the form of network control and attack. Privileges assigned to the mobile sinks can be compromised, and as a result, can be misused. Without a proper authorization, the compromised mobile sinks can control the entire network and may cause undesirable consequences. Therefore, there is a need for a proper security mechanism to authenticate the validity of mobile sinks and detect any compromised mobile sinks.

In this paper, we propose a secure multimedia data collection scheme through mobile sinks in the WMSNs. To the best of our knowledge, there is no existing solution for a secure mobile data collection in the WMSNs. The proposed approach is based on a two-layer architecture. These two layers help in securely managing the captured multimedia data. At the first layer, the identities of the participating devices, i.e., the CHs and MSNs, are verified through a lightweight four-way handshaking mechanism. At the second layer, the CHs forward the received data to the authorized mobile sinks. The major contributions are as follows.

- A lightweight four-way handshaking mechanism is applied to verify the identities of participating devices (i.e., the CHs and MSNs). It is a payload-based mutual authentication mechanism. During the authentication process, the payload is always encrypted, and the authentication mechanism incurs minimal computational overhead.
- A lightweight authentication scheme is applied to verify the identities of mobile sinks. This scheme is similar to the four-way handshaking mechanism. However, due to the mobility of sinks, hashing functions along with random numbers have been utilized to verify the identities of the mobile sinks. This authentication is first performed between the mobile sinks and BS, and then between the mobile sinks and CHs.
- The performance of the proposed scheme is analyzed in terms of various performance metrics. The simulation results prove the efficiency of our proposed scheme as compared to other well-known state-of-the-art approaches.

The rest of this paper is structured as follows. A literature review on the recent work is provided in Section II. The system model is described in Section III followed by an explanation of the proposed approach in Section IV. The experimental setup and simulation results are discussed in Section V. Finally, the paper is concluded in Section VI.

2 Literature Review

We distribute this section into two subsections. In the first subsection, we discuss various node authentication schemes, proposed for the cluster-based WSNs/WMSNs. In the second subsections, we provide an overview of various authentication schemes for mobile sinks.

2.1 Node Authentication

In [6], various user authentication schemes designed for WSNs were reviewed. In this survey, the WSNs in unattended environments are targeted, and secure and dependable user authentication mechanisms are analyzed for future research proposals, challenges, and applications. In [7], a study on an adaptive security design for malicious node detection in the cluster-based sensor networks was presented. In this study, the adaptive security modules are analyzed to improve the secure communication in the cluster-based WSNs by protecting them from external malicious nodes, using the authentication schemes.

In [8], a cluster-based node authentication scheme was proposed for the WMSNs. In this scheme, the clusters are formed through a lightweight mutual authentication mechanism between the MSNs and CHs. In [9], a geographic secured routing mechanism was used to authenticate nodes and messages in the WMSNs. In this mechanism, the Secure Hash Algorithm v3 (SHA-3) is used to minimize computational complexity for the authentication purposes. In [10], a user-friendly hybrid authentication scheme was proposed for the WSNs. In this scheme, mutual authentication and key agreement schemes are combined, using chaotic maps to minimize the computational complexity and protect user anonymity and secret passwords. In [11], a compressive sensing based approach for a secure data collection in the WSNs was proposed. In this approach, asymmetric semi-homomorphic encryption is used to ensure the privacy during the data collection, and sparse compressive matrix is used to reduce the computational costs.

2.2 Mobile Sink Authentication

In [12], an authentication scheme was proposed to detect the mobile sink replication attacks in unattended sensor networks. This scheme is based on a model where a mobile sink can collect data from a sensor node, only if it establishes a shared key with the sensor node, and passes through authentication of at least m neighbors to improve the underlying network's resilience against the replication attacks. In [13], [14], a seamless key agreement framework was proposed for the mobile sinks in an IoT-based cloud-centric secured sensor network. In this framework, a seamless secured authentication and key agreement approach, using bilinear pairing and elliptic-curve crypto-systems, is adopted to prevent node-capture attacks and support data confidentiality, mutual authentication, session-key agreement, user anonymity, password guessing, and key impersonation. In [15], a comprehensive approach for data collection was proposed for sensor-cloud systems. In this approach, the trustworthiness of mobile sinks and sensor nodes are evaluated against various type of trusts. In [16], pairing-free ID-based identification scheme was proposed to minimize computation and communication costs for registering mobile sinks in WSNs. In this scheme, the performance is evaluated in terms of computation cost and energy consumption on well-known WSN hardware platforms. In [17], distributed

and decentralized schemes were proposed to detect replication attacks in mobile WSNs. In these schemes, malicious nodes along with collaborating imposters are detected and quarantined through total number of false-positives. In [18], a lightweight algorithm was proposed to detect mobile Sybil nodes in the mobile WSNs. In this algorithm, Watchdog Nodes first technique is used to label mobile nodes, based on their movements. As a result, the Sybil nodes can be detected with a high accuracy, according to their movements and the assigned labels.

3 System Model

The system model focuses on multimedia data collection from the secured WMSNs. The MSNs capture/sense unusual events, and store the captured/sensed in their buffers. Later, the captured/sensed data are forwarded to the elected CHs. These CHs are responsible for sharing the received data with the nearby mobile sinks. The mobile sinks are then responsible to upload the data to the cloud/local servers through the Internet. As the focus is on the secure collection of multimedia data, therefore, the computational overhead of securing the underlying network needs to be the minimum.

3.1 Secure Cluster Formation Model

The secure cluster formation model is based on our previous work, published in [19], [20]. Notations used in our proposed model are illustrated in Table 1. In this model, the MSNs are denoted by αs, the CHs are denoted by rs, and the BS is denoted by a γ. Identity verification messages are exchanged between αs and rs. The identity verification messages contain various challenges that need to be decrypted by communicating entities. In our proposed scheme, the authentication process consists of four phases, i.e., session initiation, CH challenge, MSN response and challenge, and CH response. To encrypt the payload of the verification messages, Advanced Encryption Standard 128-bits (AES-128) is used. Each MSN possesses a pre-shared secret (i.e., λ) and a token (i.e., τ). The secret is used during the authentication process, and the token is used for a secure exchange of nomination and acknowledgment packets between the CHs and BS.

Notation	Description
α	Mobile sensor node (MSN)
r	CH (CH)
γ	Base station
λ	Secret
τ	Token
i	ID of mobile sensor node
j	ID of CH
η	Pseudo-random nonce
μ	Session key
ϑ	Intermediate value
C	Challenge

TABLE 1: Notations of cluster formation model

During the first phase, i.e., the session initiation, each MSN creates a joining-request message and forwards it to a targeted CH. The payload of each joining-request message contains the ID of the MSN (i.e., α_i), where $i \in \{1, 2, 3, ..., I\}$, and the header contains the ID of the targeted CH (i.e., r_j), where $j \in \{1, 2, 3, ..., J\}$).

After the session initiation, next phase is the CH challenge. In this phase, the r_j searches for a matching λ_i for the α_i, and replies back with a challenge. The challenge consists of a pseudo-random nonce (i.e., η_r), and a session key (i.e., μ) generated by an r_j. The challenge is created using the following equations.

$$\vartheta = \lambda_i \oplus \mu, \tag{1}$$

$$C_r = AES\{\lambda_i, (\vartheta \mid \eta_r)\}, \tag{2}$$

where ϑ is an intermediate value generated by an r_j, and C_r is the challenge created for an α_i.

In the MSN response phase, the α_i retrieves η_r and μ from C_r, and creates a challenge of its own, for an r_j, using the following equations.

$$\overline{\vartheta} = \eta_r \oplus \lambda_i, \tag{3}$$

$$C_i = AES\{\mu, (\overline{\vartheta} \mid \eta_i)\}, \tag{4}$$

where η_i is the pseudo-random nonce, $\overline{\vartheta}$ is the intermediate value generated by an α_i, and C_i is the challenge created for an r_j.

Upon reception, the r_j tries to retrieve η_r from α_i's challenge. If present, the status of α_i changes to $Authenticated$, and the r_j replies back with a response to α_i's challenge to complete the authentication process using the following equation.

$$C_r = AES\{\lambda_i, (\eta_i \mid \mu)\}. \tag{5}$$

Upon receiving the response computed by Eq. 5, the α_i tries to extract the η_i. If present, the status of r_j changes to $Authenticated$.

3.2 Mobile Sink Verification Model

The notations used in this model are illustrated in Table 2. The mobile sinks (i.e., βs) communicate with CHs to collect the captured data. However, each mobile sink first needs to register itself with the BS. Once registered, the mobile sinks are able to communicate with the surrounding CHs. Before contacting the BS for registration, each mobile sink generates a session key (i.e., μ_k), where k is the ID of the requesting mobile sink, and $k \in \{1, 2, 3, ..., K\}$, and a random integer x, where $x \in Z$. In this registration process, two hashing functions, i.e., H_1 and H_2, are utilized. Each β_k generates a secure request message (i.e., Θ_k) for registration using H_2, as illustrated by the following equation.

$$\Theta_k = AES\{k, (H_2(x \oplus \mu_k))\}. \tag{6}$$

The generated request is forwarded to the BS. After receiving the request message, the BS generates an encrypted challenge (i.e., C_γ) consisting of four values using the following equations and forwards it to the β_k.

Notation	Description
β	Mobile sink
k	ID of mobile sink
w, x, y, z	Random integers
Z	Set of integers
H_1, H_2	Hashing functions
Θ	Secure request message
Δ	Registration certificate
δ	Time-stamp
Ψ	Authentication value
$\overline{\mu}$	New session key
f	Bilinear group mapping function
$\hat{c}$	Available capacity of mobile sink
t	time
d	Communication range of mobile sink
v	Moving speed of mobile sink
s	Service rate of mobile sink

TABLE 2: Notations of mobile sink verification model

$$\Delta_k = w.H_1(k \mid\mid H_2(x \oplus \mu_k)), \tag{7a}$$

$$\delta_k = H_2(H_2(k \mid\mid y)), \tag{7b}$$

$$\Psi_k = H_2(k \mid\mid x \mid\mid y), \tag{7c}$$

$$C_\gamma = AES\{\Delta_k, \delta_k, \Psi_k\}, \tag{7d}$$

where $w \in Z$, $y \in Z$, Δ_k is a registration certificate, δ_k is a time-stamp, and Ψ_k is an authentication value for the β_k.

Upon reception, each β_k creates an encrypted response (i.e., C_k) using a random value z, where $z \in Z$, and replies back to complete the registration process by using the following equations.

$$a_k = H_2(\Psi_k \mid\mid H_2(x \oplus \mu_k) \mid\mid z), \tag{8a}$$

$$C_k = AES\{a_k, z\}. \tag{8b}$$

Upon receiving the response, the BS verifies identity of the β_k using **Eq. 8a**. If the equation holds, then the β_k is considered an authentic node. The β_k generates an encrypted connection request (i.e., $\overline{C_k}$) including a new session key (i.e., $\overline{\mu_k}$) by using the following equations and forwards it to the authentic r_j.

$$b_k = f(\Delta_k.H_2(\delta_k).H_1(j)), \tag{9a}$$

$$\overline{\mu_k} = f(H_2(\delta_k).\Delta_k, H_2(\delta_k).H_1(j)), \tag{9b}$$

$$\overline{C_k} = AES\{b_k, \overline{\mu_k}\}, \tag{9c}$$

where $f()$ is a bilinear group mapping function.

Upon reception, the r_j verifies the identity of β_k by using Eq. 9a. If the equation holds, the β_k is considered an authentic node. Once authenticated, the r_j can start communicating with β_k using $\overline{\mu_k}$.

Each β_k has a specific capacity, and cannot collect multimedia data from all the nearby rs. It is possible that a β_k is communicating with more than one r. In this situation, it needs to assign a specific capacity to an r_j. The available capacity (i.e., $\hat{c}$) of a β_k at time t is estimated by the following equations.

$$t = \frac{d}{v}, \tag{10}$$

$$\hat{c} = s \times t, \tag{11}$$

where, d is communication range, v is moving speed, and s is service rate of a β_k.

4 Secured and Authorized Data Collection Scheme

In this section, we explain our proposed Secured and authorized Scheme (SaaS) for mobile multimedia data collection. The block diagram depicting the traditional mechanism of the WMSNs is shown in Fig. 1. The block diagram depicting the operational mechanism of the proposed SaaS is shown in Fig. 2. The proposed SaaS is divided into two layers, i.e., the device and mobile sink layers. The clusters are formed at the device layer via a lightweight handshaking mechanism, while the data is collected from the CHs using the mobile sinks at the mobile sink layer.

4.1 Secure Cluster Formation

This phase is divided into two sub-phases, i.e., the election of the CHs, and mutual authentication between the MSNs and CHs. The CH election is a token-based approach which enables the MSNs with highest energy levels to become the CHs. The remaining nodes join the CHs to form clusters and forward data to the elected CHs.

4.1.1 CH Election

In our proposed SaaS, the WMSN consists of 500 randomly deployed MSNs. The MSNs with minimum energy levels become members in a cluster while the remaining nodes become the CHs. The elected CHs are responsible for authentication of member nodes, and data reception and forwarding to the BS. Each MSN possesses a λ and a τ. The τ is used to exchange control packets i.e., nomination packets (i.e., p), and acknowledgment packets (i.e., ACK), between the CHs and BS. On the other hand, the λ is used to perform mutual authentication inside the clusters between the MSNs and CHs.

At the beginning of each round, each α_i broadcasts a p_i to the BS. Each p_i contains the α's ID (i.e., i), secret (i.e., λ_i), and residual energy (i.e., α_e). Upon reception, the BS extracts i and α_e from p, and computes the average energy threshold (i.e., E), by using the following equation.

$$E = \sum_{i=1}^{N} \frac{\alpha_e}{N}, \tag{12}$$

where N represents the total number of MSNs.

An α_i having energy equal or greater than E is considered eligible to be elected as a CH. An exception may occur at the start of each round when all the MSNs have almost the same energy levels. In this case, the election of a CH is based on minor differences in the energy values and can go up to four decimal digits. In upcoming simulation rounds,

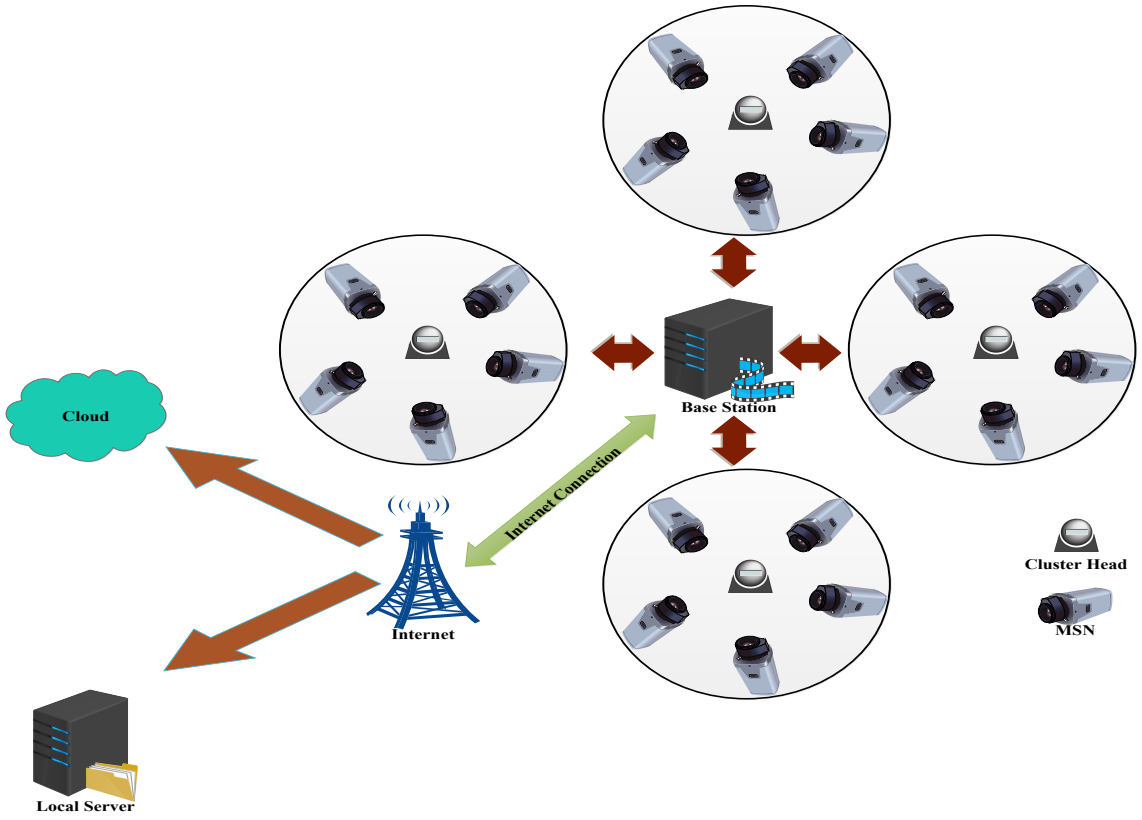


Fig. 1: Traditional Mechanism of WMSN

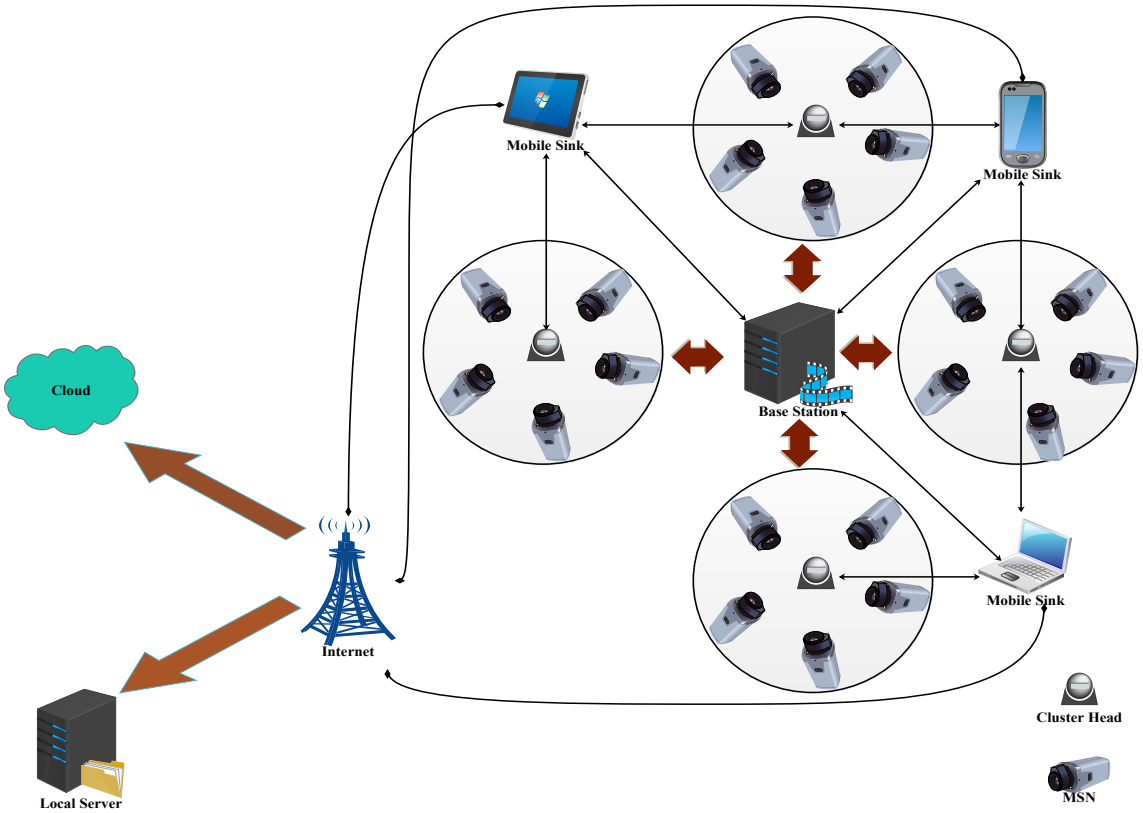


Fig. 2: Operational Mechanism of SaaS

it is possible to have more than one nominee for a CH role. In this case, a CH is selected based on the following rules.

- α_e should be equal or greater than E.
- Node α_i is not elected as a CH in past rounds.
- Nominees in the same geographical location are evaluated based on their previous history of election if they have the same energy levels.

After the election, the BS broadcasts a p_γ to the elected CHs. This p_γ contains the IDs of the elected CHs and their neighboring MSNs. 16-bit tokens similar to the tokens of the elected CHs are also appended in the header of each p_γ. Upon reception, an elected r_j matches the received token with its own. If matches, three operations are performed at the elected r_j, 1) retrieval of the appended IDs of the neighboring MSNs and storing in an array (i.e., B[α_i][λ_i]), 2) creation of an ACK packet that includes its own token and forwards it back to the BS, and 3) advertising itself to its neighboring MSNs. It is possible that an MSN receives advertisements from more than one CH. In this case, an MSN associates itself with the nearest one by sending a joining request. The location of an MSN with respect to a broadcasting r_j is computed by the Euclidean distance (i.e., $\hat{d}$), as shown in the following equation.

$$\hat{d} = \sqrt{(x - x_i)^2 - (y - y_i)^2}. \tag{13}$$

The joining process requires mutual authentication which is explained in the next subsection.

4.1.2 Mutual Authentication

Upon successful advertisement, each MSN tries to become a part/member of a cluster, led by a CH. The aim of cluster formation is to secure data transmission from the MSNs to the CHs that are authorized to share the data with either the BS or mobile sinks. To authenticate an MSN, a payload-based handshaking mechanism is used. The 128-bit AES encryption algorithm is used to encrypt the messages shared between the nodes. Unlike AES-192 and AES-256 algorithms, the AES-128 algorithm requires less computational resources and is suitable for real-time data processing [21]. The MSNs exchange multiple handshake messages with the CHs as shown in Fig. 3. Once authenticated, an MSN is allowed to forward the data to the nearest CH, to whom it belongs.

In the session initiation phase, an MSN sends a join-request to the nearest CH. The join request contains j, i and λ_i. The IDs are 8-bits long, making the join request messages extremely lightweight. A maximum of four join requests can be sent to a CH. After four attempts, any further attempt is simply ignored. In the join-request, an i is encapsulated in the payload while a j is encapsulated in the header. A frame check sequence is also appended as a trailer of the payload for error detection and correction.

Upon receiving the join-request, the CH retrieves j and i from the header and payload fields, respectively. If the CH's ID does not match with r_j, the request is simply discarded otherwise the CH proceeds with the CH challenge task. The CH also searches for a matching λ_i. If a match is found, the requesting device is considered authentic, and a challenge is

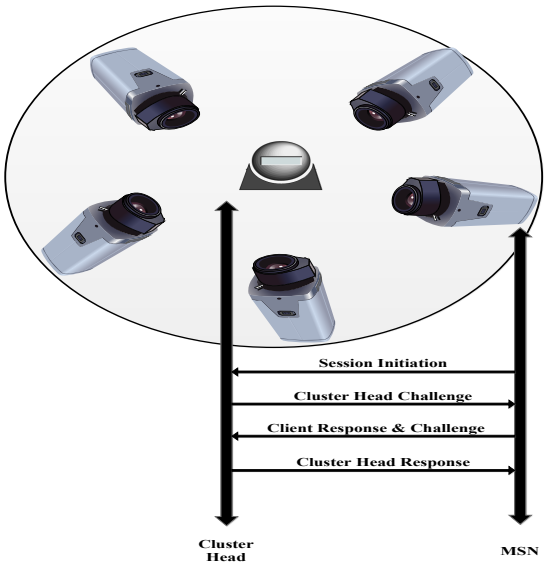


Fig. 3: Secured Cluster Formation via Handshaking

forwarded back as a response. The challenge consists of an encrypted payload and is computed by Eq. 1-2.

In the MSN response phase, the MSN needs to decipher the encrypted payload to extract μ and η_r. Upon successful deciphering, the MSN is authenticated and able to create a challenge, using Eq. 3-4, to verify the identity of the CH.

Upon receiving the MSN's challenge, the CH deciphers the challenge to observe an η_r. If present, it retrieves an η_i, and creates a 256-bit encrypted payload, using Eq. 5, to transmit as a response to MSN's challenge and changes the status of MSN to $Authenticated$.

After receiving the CH's response, the MSN verifies the identity of an r_j. Upon successful verification, the mutual authentication process is completed, and the MSN is now permitted to forward the data to the selected r_j. Furthermore, the selected r_j is now authorized to share the data collected with the mobile sinks. The entire mutual authentication process is summarized in Algorithm 1.

4.2 Mobile Sink Verification

In the cluster-based data communication, the CHs collect the data from member MSNs, and forward to the BS. In our proposed SaaS, we are assuming that the Internet connection is down, and the BS is unable to forward the data to the local/cloud servers. In this particular scenario, the mobile sinks can collect the data from the CHs, and forward to the local/cloud servers. However, the use of mobile sinks raises network security issues. To ensure a secure data communication from the CHs to mobile sinks, there is a need to authenticate the mobile sinks. In the following subsections, we explain registration and authentication of the mobile sinks in a WMSN.

4.2.1 Mobile Sink Registration

For a secure data exchange between the CHs and mobile sinks, there is a need to register the mobile sinks with the BS as shown in Fig. 4. In real-world scenarios, the mobile sinks are always on the move. As a result, the CHs get a very

Algorithm 1 Payload-based Mutual Authentication

Initialization:

- $\alpha \longleftarrow [i, \lambda_i]$
- $r \longleftarrow [j, \lambda_j]$
- Input [λ_i=λ_j=2^{128}]

$(\alpha_i, \lambda_i, r_j) \rightarrow r_j$
if r_j == r_j **then**
  if [α_i,λ_j]==A[α_i][λ_i] **then**
    $\alpha_i \longleftrightarrow r_j$ ▷ Session created between α_i and r_j
    $r_j \rightarrow \alpha_i : \{C_r$, challenge created$\}$
  else
    α_i is unauthorized
  end if
else
  The request is for a different r and discarded
end if
$\alpha_i \longleftarrow \{\lambda_i, (\vartheta \mid \eta_r)\}$ ▷ Decipher challenge
$\alpha_i \rightarrow r_j : \{C_i,\ \alpha_i$ respond with a challenge$\}$
if η_r exists **then**
  Access granted
  α_i is authenticated
  $r_j \rightarrow \alpha_i : \{C_r, r_j$ responds back with a new challenge$\}$
else
  Access denied
  α_i is unauthenticated
end if
if η_i exists **then**
  r_j is authenticated
  Data can be exchanged now
else
  r_j is unauthenticated
end if

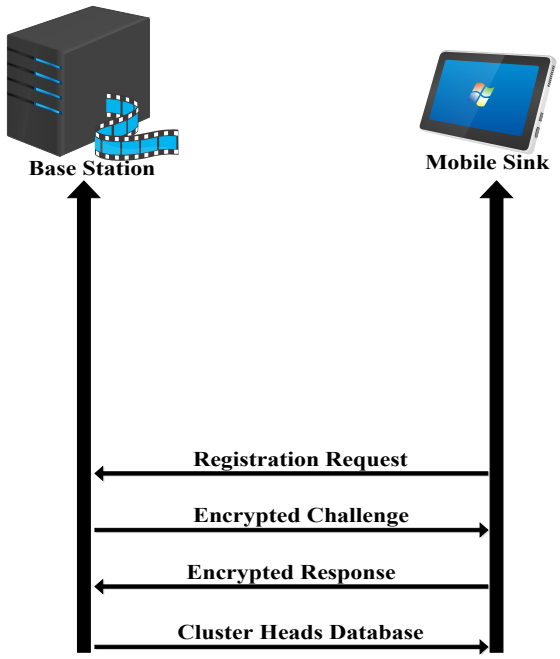


Fig. 4: Registration of Mobile Sink with Base Station

short amount of time to share the data with the authorized mobile sinks. In the proposed SaaS, each β_k generates an μ_k, and some random integers, i.e., x and z. The generated μ_k and random integers are used to register a β_k with the BS, and these numbers need to be generated before forwarding a registration request. The registration process is based on two hashing functions, i.e., H_1 and H_2, for mapping and secure hashing purposes, respectively [22]. The first hashing function, i.e., H_1, is used to perform a map to point hashing operation, and the second hashing function, i.e., H_2, is used to perform one-way secure hashing. For both hashing operations, the range of values is between $\{0, 1\}$.

After generating the random integers and μ_k, the first step in the registration process is to generate and send an encrypted registration request, i.e., Θ_k, to the BS. This registration request consists of k, μ_k, and one of the generated random integers, i.e., x. The registration request is generated using Eq. 6.

After receiving the encrypted registration request, the BS decrypts it and extracts all the embedded values. The BS also generates two random integer values, i.e., w and y. The BS uses the extracted values to generate an encrypted challenge. The encrypted challenge consists of three different values, i.e., a registration certificate (i.e., Δ_k), a time-stamp (i.e., δ_k), and an authentication value (i.e., Ψ_k). The BS generates two copies of the registration certificate. One copy of the registration certificate is stored at the BS for future communication, and the other copy is used in the encrypted challenge. The time-stamp represents the total amount of time a mobile sink can spend in the WMSN to collect the data from the CHs. The authentication value is used to create a response to the encrypted challenge at the mobile sink. The three different values for the encrypted challenge are computed using Eq. 7a-7c. After computing these values, the BS embeds them into an encrypted payload, and sends as a challenge to the requesting β_k, using Eq. 7d.

After receiving the encrypted challenge, the requesting β_k decrypts the payload to extract the embedded values. The registration certificate and time-stamp are stored at the mobile sink, and the authentication value is used to create a response to the BS's challenge. The mobile sink uses the random integer values x and z along with the extracted authentication value and session key to create an intermediate value using Eq. 8a. The generated intermediate value along with the random integer value z is encrypted in the payload of the message, using Eq. 8b, and is sent to the BS as a response to the challenge.

After receiving the encrypted response, the BS verifies the identity of the mobile sink, using Eq. 8a. If the equation holds, the mobile sink is considered authentic, otherwise, it is considered an intruder node. After verifying the identity of the mobile sink, the BS performs three operations, 1) shares the registration certificate and time-stamp of the verified mobile sink with all the CHs, 2) forwards the database of elected CHs to the verified mobile sinks, and 3) forwards the location information of the nearby CHs to the verified mobile sinks to complete the registration process.

4.2.2 Mobile Sink Authentication

After receiving the database of all the elected CHs, the mobile sink stores the database in its buffer. Secondly, it broadcasts a request to all the nearby CHs for establishing a connection to collect the data. In response, the nearby CHs reply back by sharing their unique IDs, i.e., j. If the IDs exist in the stored database, the mobile sink generates two intermediate values using the IDs of the CHs and its own time-stamp, as shown in Eq. 9a-9b. These generated intermediate values are encapsulated into the payload of a connection request message, as shown in Eq. 9c, and sent to the CHs. Upon receiving, the CHs verify the identity of the mobile sink, using Eq. 9a. If the equation holds, the mobile sink is considered an authentic node. Once the mobile sink is authenticated, the CHs start communication with it.

It is possible that the mobile sink is connected to more than one CH at a time. In this case, it broadcasts a data collection request to all the communicating/connected CHs. The data is arranged in their buffers, based on priority levels. Due to the large size, the multimedia data is always of high priority, and need to be transmitted first. The data collection requests provide information about the size of data. Based on the provided information, the mobile sink assigns capacities to the CHs.

Once the mobile sink broadcasts a request for a data collection, the nearby CHs reply back with capacity allocation requests, based on their low and high priority data. The mobile sink first processes high priority data requests and assigns high capacities to the requesting CHs. Once the capacity is assigned, the CHs immediately start transmitting the data. If there is any capacity left at the mobile sink, it is evenly assigned to the CHs with low priority data requests. The capacity allocation process is a continuous process and stops when no more capacity is left. Furthermore, the capacity of the mobile sink is always updated according to its moving speed. The current available capacity of the mobile sink at time t is estimated by using Eq. 10-11. It is possible that there are multiple mobile sinks moving around a CH. In this case, a mobile sink is selected based on the maximum available capacity. The mobile sink registration, authentication, and data exchange processes are summarized in Algorithm 2.

Algorithm 2 Mobile Sink Verification

Initialization:

- $\beta \longleftarrow k$
- $\triangleright$ where k represents number of βs in the network
- $\beta \longleftarrow [\mu, x, z]$
- $\gamma \longleftarrow [w, y]$

$\beta_k \rightarrow \gamma : \{\Theta_k$, secured request message$\}$
$\gamma \rightarrow \beta_k : \{C_\gamma$, encrypted challenge$\}$
$\beta_k \rightarrow \gamma : \{C_k$, encrypted response$\}$
if Ψ_k exists **then**
    $\triangleright$ γ checks for the presence of Ψ_k in C_k
    β_k is registered
    $\gamma \rightarrow rs : \{\Delta_k$ and δ_k, these values are shared with rs$\}$
    $\gamma \rightarrow \beta_k : \{rs$, share database and locations of rs with $\beta_k\}$
else
    β_k is unregistered
end if
if j exists **then**
    Send a connection request
else
    r_j is unauthenticated
end if
if k exists **then**
    β_k is authorized
    $\beta_k \rightarrow r : \{\hat{c}$, assigning available capacity to r$\}$
    if $P_{r[a]} > P_{r[b]}$ **then**
        β_k assigns $\hat{c}$ to $r[a]$ $\triangleright \forall\ a, b \in j$
        $\triangleright$ P represents priority-level
        $\beta_k \longleftarrow r[a] : \{D, r[a]$ start transmitting high-priority multimedia data $D\}$
    else
        β_k assigns $\hat{c}$ to $r[b]$
    end if
else
    β_k is unauthenticated
end if

5 Experimental Setup and Simulation Results

In this section, first, we discuss experimental setup. Later, the performance of our proposed SaaS is evaluated in detail.

5.1 Network Setup

For simulations, we use Matlab 2017a. The network consists of 500 randomly deployed MSNs, out of which 5% of the nodes are selected as the CHs. All the nodes are deployed in a $500 \times 500m^2$ area. The mobile sinks and MSNs communicate over an IEEE 802.11e network. The communication range of each MSN is set to $100m$. The mobile sinks are always on the move with a speed of v using random waypoint mobility model [23]. We do not consider any pause time in their movements.

5.2 Performance Evaluation

In this subsection, we evaluate the performance of our proposed SaaS approach against two similar state-of-the-art schemes, i.e., Secured Query Processing Scheme (SQPS) [24], and Modified Secured Query Processing Scheme (MSQPS) [25]. Both these schemes use a cluster-based hierarchical approach for securing the exchange of data. First, we compare the proposed SaaS against the SQPS and MSQPS in terms of various security primitives followed by their robustness against various attacks, handshake duration, authentication rate, data freshness, and packet delivery ratio, respectively.

In the proposed SaaS, the node authentication phase is a two-step process. First, the CHs are elected by the BS. Later, a mutual handshaking mechanism is performed to authenticate both the CHs and MSNs for data collection. The proposed SaaS uses a Message Authentication Code (MAC) in Cipher Block Chaining (CBC) mode to ensure the authentication, as shown in Table 3. The SQPS and

MSQPS, on the other hand, use the simple MAC for authentication. To ensure confidentiality, the proposed SaaS uses a payload-based symmetric encryption, that is extremely lightweight. In comparison, the SQPS and MSQPS employ RC5-based symmetric cryptography to maintain the confidentiality. For data freshness, the proposed SaaS uses pseudo-random nonces (i.e., η_i and η_r). These nonces have a one-time usage during the handshake mechanism and are non-reproducible. This feature of these nonces ensures that redundant packets are not replayed by malicious nodes, and the freshness of genuine data packets remain intact. The SQPS and MSQPS use Query Response (QR) packets to ensure the data freshness. In this case, each member node of a cluster transmits a 4-tuple packet to its respective CH. Each QR packet ensures the freshness at the expense of a high computational cost at the CH.

TABLE 3: Security Analysis

Security Services	SQPS	MSQPS	SaaS
Authentication	MAC	MAC	CBC
Confidentiality	Symmetric	Symmetric	Symmetric
Freshness	QR	QR	Nonces

We evaluate the resilience of the three schemes against various malicious attacks as shown in Table 4. In the proposed SaaS, the presence of η_i, η_r and λ_i, protects the underlying network from different types of malicious attacks. Random combinations of η_i and η_j in the proposed SaaS make it extremely difficult for an adversary to replay packets. The SQPS and MSQPS use QR packets to prevent the replay of malicious packets. In the proposed SaaS, the use of λ_i restricts any unauthorized node from establishing connections with the CHs. As a result, it is extremely difficult for the malicious nodes to launch resource exhaustion and DoS attacks. As λ_i is a hard-coded secret, any attempt for tempering can easily be identified, and an alarm can be triggered. This feature of λ_i ensures that a malicious node does not clone itself by stealing the identity of a legitimate node. The BS has prior knowledge about the identities of all the legitimate nodes that prevent the malicious nodes from launching the Sybil attacks, as they are unable to impersonate themselves. The SQPS and MSQPS, on the other hand, are unable to detect the resource exhaustion, DoS, cloning and Sybil attacks.

TABLE 4: Resilience against Attacks

Attacks	SQPS	MSQPS	SaaS
Replay	Yes	Yes	Yes
Resource Exhaustion	No	No	Yes
Denial-of-Service	No	No	Yes
Cloning	No	No	Yes
Sybil	No	No	Yes

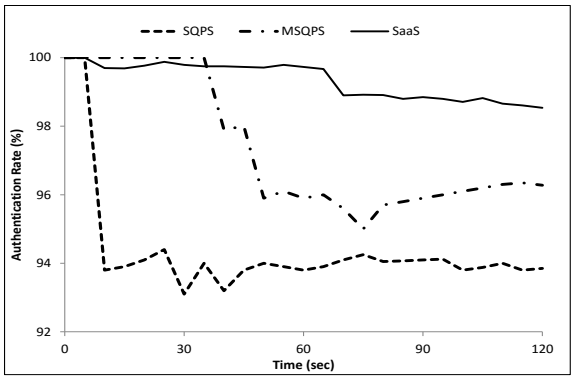
We evaluate the performance of the proposed SaaS against the SQPS and MSQPS in term of handshake duration for various cluster sizes as shown in Table 5. The handshake duration depends on the total number of member nodes in a cluster and is usually measured in milliseconds (ms). The handshake approach of the proposed SaaS takes less time as compared to the SQPS and MSQPS. Unlike these approaches, the CHs in our proposed SaaS use local knowledge of neighboring nodes for handshaking and authentication purposes. As the table shows, increasing the size of a cluster has a relatively smaller impact on the handshake duration in the proposed SaaS. However, the handshake duration increases significantly in the case of the SQPS and MSQPS with an increase in the cluster size. This is mainly due to the fact that these schemes use complex symmetric encryption suites that require excessive processing at each CH.

TABLE 5: Handshake Duration (ms)

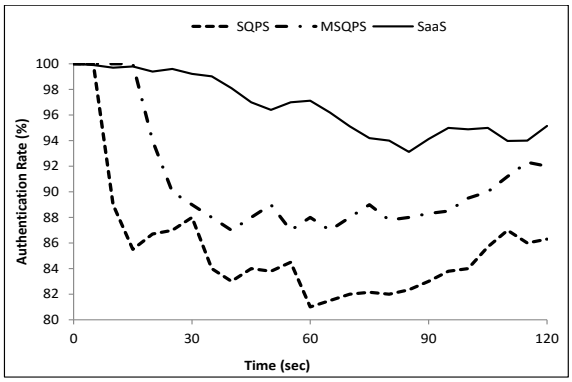
Cluster Size	SQPS	MSQPS	SaaS
15	4.03	3.7	2.08
20	4.99	4.1	2.16
25	5.31	4.33	2.37
40	7.11	5.91	2.99

In Fig. 5, the authentication rate for received messages is shown. As the figure shows, the authentication rate decreases with an increase in the number of malicious nodes in the underlying network. This is mainly because the CHs have to process requests from multiple sensor nodes (MSNs in the case of the proposed SaaS) at a given time, which allows one or more malicious nodes to sneak through the underlying authentication mechanism. Another reason for this decrease in the authentication rate is as the number of malicious nodes increases, the number of attempts by these nodes to transmit their unauthenticated information increases. In Fig. 5(a), we compare the proposed SaaS against the MSQPS and SQPS in the presence of 25 malicious nodes. Although our proposed SaaS scheme drops slightly for the first 10 seconds, it sustains itself by ensuring a higher authentication rate. Minimum authentication rate is 98.5% that shows that the proposed SaaS is highly robust against the detection of malicious nodes. In comparison, as the time passes, the authentication rates of MSQPS and SQPS drop to 96.28% and 93.85%, respectively. In the presence of 50 malicious nodes, the proposed SaaS has a minimum authentication rate of 93.1%, as shown in Fig. 5(b). In comparison, the authentication rates of MSQPS and SQPS drop to as low as 87.8% and 81.2%, respectively. Fig 5 concludes that the proposed SaaS is highly efficient in ensuring the message integrity in comparison to the MSQPS and SQPS because the authentication rate is an indicator of message integrity.

In Fig. 6, the proposed SaaS is compared with the MSQPS and SQPS in term of data freshness. A higher percentage of data freshness means a scheme is more resilient against the replay attacks. In the presence of 25 malicious nodes, the average data freshness of the proposed SaaS is 99.45%, MSQPS is 97.28% and SQPS is 91.8%, respectively, over a period of 120 seconds. These results are depicted in Fig. 6(a). The proposed SaaS achieves a higher percentage of data freshness and sustains itself against the replay attacks over the course of time. Unlike the proposed SaaS, the MSQPS experiences a significant decrease after 30 seconds of the network deployment. The SQPS, on the other hand, experiences a significant drop of 11% in the data freshness, after the initial 20 seconds of the network deployment. In

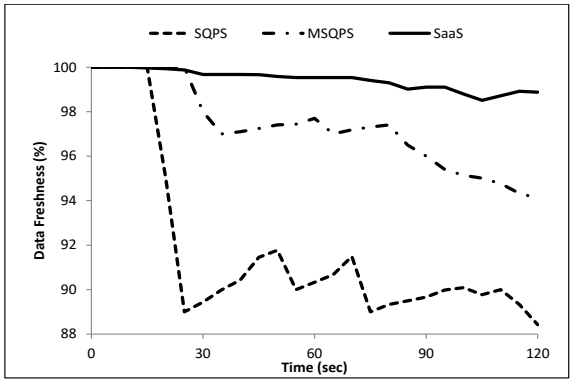


(a) 25 malicious nodes

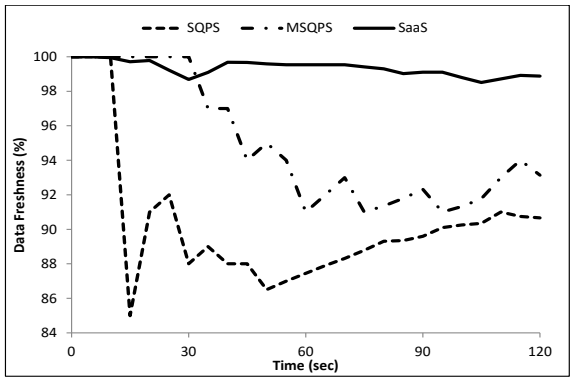


(b) 50 malicious nodes

Fig. 5: Authentication rate for varying number of malicious nodes



(a) 25 Malicious Nodes



(b) 50 Malicious Nodes

Fig. 6: Data Freshness (%) for various malicious nodes

Fig. 6(b), all the three schemes are compared in term of data freshness in the presence of 50 malicious nodes. The average data freshness of the proposed SaaS is 99.3%, MSQPS is 94.9% and SQPS is 90.3%, respectively, over the course of time. In the beginning, the proposed SaaS drops briefly beyond the MSQPS, however, the drop is only 1.1% for a short period of 10 seconds. After 30 seconds, the MSQPS drops to as low as 91% and stays low over the specified period of time. The SQPS, on the other hand, drops by 15% after the initial 20 seconds and stays low over the course of time. Fig 6 concludes that as the number of malicious nodes increases, the freshness percentage of the received data decreases. An increase in the number of malicious nodes increases the number of attempts made by such nodes to transmit their data packets. As a result, the redundant data packets are transmitted which affects the freshness metric and at the same time, increases the replay attacks.

In Fig 7, packet delivery ratio of the three schemes is shown for varying number of malicious nodes. The average packet delivery ratio for SaaS is 95%, MSQPS is 84% and SQPS is 74%, respectively. Unlike SaaS, MSQPS and SQPS experience a significant drop in packet delivery ratio with an increase in the number of malicious nodes. This parameter is associated with authentication rate, in general. During the authentication, the malicious nodes that go undetected, acquire time division multiple access (TDMA) slots from cluster heads and broadcast their fake data packets. Both MSQPS and SPQS experience a significant drop in packet delivery ratio as the number of malicious nodes increases from 20 to 100, respectively.

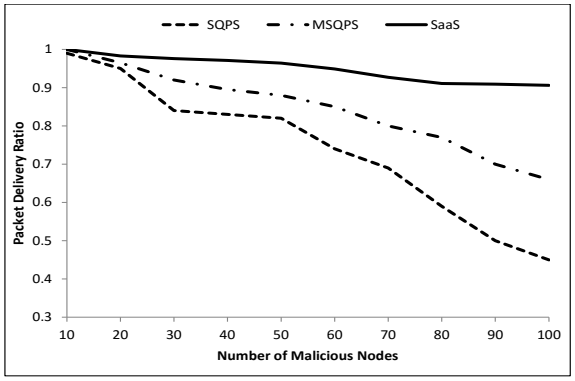


Fig. 7: Packet delivery ratio with varying number of malicious nodes

6 Conclusion

In this paper, we have proposed a node authentication scheme, called SaaS. It has been designed for multimedia data collection through mobile sinks in a wireless multimedia sensor network. The proposed SaaS has based on the clustering concept. A mutual handshaking mechanism has been applied to authenticate both the MSNs and CHs.

Once authenticated, the MSNs forward the captured data to the elected CHs, and then it is the responsibility of the CHs to share the collected data with the nearby verified mobile sinks. The mobile sink verification involves their registration with the base station and authentication with the CHs. The simulation results have shown that our proposed SaaS performs better in terms of resilience, handshake duration, authentication rate, data freshness, and packet delivery ratio when compared with other established state-of-the-art approaches. Regarding future work, we shall use the simulation results produced in this paper as a base to perform further enhancements in our proposed SaaS to support mobile MSNs and sinks with random speeds. Besides, we aim to implement the proposed scheme on real testbeds in open field environment to measure various phenomena of interests in order to analyze various QoS metrics.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, "Internet of multimedia things: Vision and challenges," *Ad Hoc Networks*, vol. 33, pp. 87–111, 2015.

[3] M. Usman, N. Yang, M. A. Jan, X. He, M. Xu, and K. Lam, "A joint framework for qos and qoe for video transmission over wireless multimedia sensor networks," *IEEE Transactions on Mobile Computing*, 2017.

[4] G. Mali and S. Misra, "Trast: Trust-based distributed topology management for wireless multimedia sensor networks," *IEEE Transactions on Computers*, vol. 65, no. 6, pp. 1978–1991, 2016.

[5] J. P. J. Peixoto and D. G. Costa, "Wireless visual sensor networks for smart city applications: A relevance-based approach for multiple sinks mobility," *Future Generation Computer Systems*, vol. 76, pp. 51–62, 2017.

[6] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Networks*, vol. 27, pp. 159–194, 2015.

[7] M. A. Jan, P. Nanda, X. He, and R. P. Liu, "A sybil attack detection scheme for a forest wildfire monitoring application," *Future Generation Computer Systems*, vol. 80, pp. 613–626, 2018.

[8] M. Usman, M. A. Jan, X. He, and P. Nanda, "Data sharing in secure multimedia wireless sensor networks," in *Trustcom/BigDataSE/ISPA, 2016 IEEE*. IEEE, 2016, pp. 590–597.

[9] B. P. Laxmi and A. Chilambuchelvan, "Gsr: Geographic secured routing using sha-3 algorithm for node and message authentication in wireless sensor networks," *Future Generation Computer Systems*, vol. 76, pp. 98–105, 2017.

[10] S. Kumari, X. Li, F. Wu, A. K. Das, H. Arshad, and M. K. Khan, "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Generation Computer Systems*, vol. 63, pp. 56–75, 2016.

[11] P. Zhang, S. Wang, K. Guo, and J. Wang, "A secure data collection scheme based on compressive sensing in wireless sensor networks," *Ad Hoc Networks*, vol. 70, pp. 73–84, 2018.

[12] S. Li, W. Wang, B. Zhou, J. Wang, Y. Cheng, and J. Wu, "A (m, m) authentication scheme against mobile sink replicated attack in unattended sensor networks," *IEEE Wireless Communications Letters*, 2017.

[13] F. Al-Turjman, Y. K. Ever, E. Ever, H. X. Nguyen, and D. B. David, "Seamless key agreement framework for mobile-sink in iot based cloud-centric secured public safety sensor networks," *IEEE Access*, vol. 5, pp. 24 617–24 631, 2017.

[14] M. Usman, M. A. Jan, and X. He, "Cryptography-based secure data storage and sharing using hevc and public clouds," *Information Sciences*, vol. 387, pp. 90–102, 2017.

[15] T. Wang, Y. Li, W. Fang, W. Xu, J. Liang, Y. Chen, and X. Liu, "A comprehensive trustworthy data collection approach in sensor-cloud system," *IEEE Transactions on Big Data*, 2018.

[16] K.-A. Shim, "Basis: A practical multi-user broadcast authentication scheme in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1545–1554, 2017.

[17] T. Dimitriou, E. A. Alrashed, M. H. Karaata, and A. Hamdan, "Imposter detection for replication attacks in mobile sensor networks," *Computer Networks*, vol. 108, pp. 210–222, 2016.

[18] M. Jamshidi, E. Zangeneh, M. Esnaashari, and M. R. Meybodi, "A lightweight algorithm for detecting mobile sybil nodes in mobile wireless sensor networks," *Computers & Electrical Engineering*, vol. 64, pp. 220–232, 2017.

[19] M. Jan, P. Nanda, M. Usman, and X. He, "Pawn: a payload-based mutual authentication scheme for wireless sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 17, 2017.

[20] M. A. Jan, F. Khan, M. Alam, and M. Usman, "A payload-based mutual authentication scheme for internet of things," *Future Generation Computer Systems*, 2017.

[21] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[22] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *International Workshop on Public Key Cryptography*. Springer, 2004, pp. 277–290.

[23] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Transactions on mobile computing*, vol. 2, no. 3, pp. 257–269, 2003.

[24] A. Ghosal, S. Halder, S. Sur, A. Dan, and S. DasBit, "Ensuring basic security and preventing replay attack in a query processing application domain in wsn," in *International Conference on Computational Science and Its Applications*. Springer, 2010, pp. 321–335.

[25] A. Ghosal and S. DasBit, "A lightweight security scheme for query processing in clustered wireless sensor networks," *Computers & Electrical Engineering*, vol. 41, pp. 240–255, 2015.

Muhammad Usman received a Ph.D. in Computer Systems from the School of Electrical and Data Engineering, University of Technology Sydney, Australia, in 2017. Currently, he is working as a postdoc research assistant in the Department of Computer Science and Software Engineering, Swinburne University of Technology, Australia. His research interests include audio, image and video processing, multimedia communication, security and privacy in wireless networks, Internet of Things, and error concealment techniques.

Mian Ahmad Jan is an Assistant Professor in the Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan. He holds a Ph.D. in Computer Systems from the University of Technology Sydney, Australia. His research interests include cluster-based hierarchical routing protocols, congestion detection and mitigation and intrusion and malicious attack detection in wireless sensor networks, Internet and web of things.

Xiangjian He received a Ph.D. in Computing Sciences from the University of Technology, Sydney, Australia, in 1999. Since 1999, he has been with the University of Technology, Sydney. He is currently a full professor and the director of Computer Vision and Pattern Recognition Laboratory there.

Jinjun Chen is a Professor from Faculty of Science, Engineering and IT, Swinburne University of Technology, Australia. He holds a PhD in Computer Science and Software Engineering (2007) from the Swinburne University of Technology, Melbourne, Australia. His research interests include cloud computing, big data, and data intensive systems. He has published more than 130 papers in high quality journals and conferences.