Efficient Authentication Algorithm for Secure Remote Access in Wireless Sensor Networks

Peter Sungu Nyakomitta*  Vincent Omollo Nyangaresi  Solomon Odhiambo Ogara

1. Faculty of Biological & Physical Sciences, Tom Mboya University College, Homabay, Kenya
2. School of Informatics and Innovation Systems, Jaramogi Oginga Odinga University of Science and Technology, Bondo, Kenya

1. Introduction

A wireless sensor network (WSN) typically consists of dynamic battery powered cooperative nodes that perceive their environment in real-time and transmit the collected data to the nearest gateway node (GWN) through wireless channels [1]. As such, the sensors, remote users and GWN are the participants in any WSN communication process [2]. Since the GWN has relatively high computational power and energy compared with the sensor nodes (SNs), they can forward the received data to remote external users located further way. Consequently, WSN offer infrastructure-free packet exchanges devoid of centralized access points. These WSNs have self-configuring ability [3], and this has endeared them to applications such as industrial automation, military surveillance and process monitoring.

According to [1], the ability of sensing and comprehending unattended environments has led to their increased adoption in various domains. However, their deployments in unattended scenarios expose WSNs to numerous attacks, including physical capture that are then utilized as vectors to mount further attacks such as side-channeling [4]. As such, it is critical that these security issues be addressed prior to their deployments [5]. The open wireless channel that is utilized to relay packets from the SNs to GWNs, and also from the GWNs to remote users exposes the broadcasted intelli-

*Corresponding Author:
Peter Sungu Nyakomitta,
Faculty of Biological & Physical Sciences, Tom Mboya University College, Homabay, Kenya;
Email: pnyakomitta@yahoo.com

KEYWORDS: Attacks, Authentication, ECC, Key agreement, Privacy, Security, WSN
gence to many privacy and security risks [4]. This may include malicious packets injections, eavesdropping, packet re-direction, modifications among others.

As explained in [7], the heterogeneity of communication protocols deployed in WSN result in network clustering whose cooperation is limited to low caliber message exchanges. This renders the design and application of global security solutions in these deployments a bit cumbersome. Although 5G may facilitate WSN automation as well as programmability through the incorporation of Software-Defined Networks (SDN), the protection of packets exchanged over the control and data planes is still crucial [8].

Considering lower layer security at the link and network layers, techniques such as internet protocol security (IPsec) and internet key exchange (IKE) are normally deployed. However, the SNs have limited energy and computational power to handle both IPsec and IKE [9]. There is therefore a need to design lightweight mutual authentication algorithms for both lower layer and upper layer communication protection. The main contributions of this paper include the following:

- An algorithm that effectively authenticates a remote user to the sensor nodes is developed to protect against WSN adversarial attacks. It is only after successful mutual authentication that remote users can access sensor data.
- A session key is derived for protecting exchanged packets over the insecure gateway node-sensor node and gateway node-remote user wireless channels.
- Real device and user identities are enciphered using secret and public keys to thwart any spoofing attacks.
- Security analysis shows that the proposed algorithm offers perfect forward key secrecy is robust against side-channel, traceability, offline guessing, replay and impersonation attacks.

The rest of this article is organized as follows: section 2 presents some past research in this research domain, while section 3 provides an outline of the system model. On the other hand, section 4 presents and discusses the obtained results, while section 5 concludes the paper and offers some future work in this area.

2. Related Work

The rich application domains for WSN have led to numerous schemes aimed at the protection of the exchanged packets. For instance, authors in [10] have proposed an IP based scheme while a location based protocol has been presented in [11]. However, the techniques in [10] and [11] result in increased network latency. On the other hand, the elliptic curve cryptography (ECC) based scheme presented in [12] is vulnerable to side-channel, traceability and offline-guessing attacks. Similarly, an ECC based three factor authentication algorithm has been presented in [13], but fails to offer protection against privileged insider attacks. A lightweight two-factor authentication scheme has been introduced in [14], but which is vulnerable to forgery, identity and password guessing attacks. Although the protocol in [15] offers three factor authentication and key agreement, it cannot provide backward key secrecy, and is susceptible to both known session ephemeral and offline password attacks. On the other hand, the algorithm in [16] is susceptible to side-channel and offline guessing attacks.

Fuzzy logic and biometric based protocol has been developed in [17] to offer three factor authentication in WSN. However, this scheme cannot offer forward key secrecy and is susceptible to side-channel, offline password guessing, stolen smart card and stolen verifier attacks. The symmetric key based protocol is presented in [18] while a three factor authentication algorithm is introduced in [19]. However, the techniques in [18] and [19] are susceptible to offline password guessing and impersonation attacks, and cannot uphold forward key security [20]. On the other hand, the fuzzy verifier based technique presented in [21] is not robust against replay attacks. Authors in [22] have presented a two factor authentication scheme while the techniques in [23] and [24] both deploy user biometric for authentication. Although, the schemes in [22-24] have reduced authentication latencies, they have increased complexities.

Authors in [25-27] have introduced bilinear pairing based mutual authentication schemes, but which results in excessive computational overheads [28]. On the other hand, the smart card based biometric authentication algorithm in [29] cannot provide anonymity and is vulnerable to impersonation attacks [15]. An authenticated key agreement technique is developed in [30], but which is susceptible to known session ephemeral, offline password and impersonation attacks [31]. The WSN intrusion scheme presented in [32] has high false alarm rate while the protocol introduced in [31] is susceptible to traceability and smart card loss attacks [33].

Machine learning based techniques for intrusion detection in WSN have been developed in [34-37] based on neural networks, support vector machine, multi-layer perceptron, and neural networks with watermarking. While these algorithms improve the accuracy of network anomaly detection models, they also introduce high computational cost which is inadequate for WSNs. Although these techniques boost detection accuracy, they result in high computation complexities. On the other hand, the algorithm introduced in [33] for three factor authentication is vulnerable to privileged insider attacks.

3. System Model

The network architecture in the proposed algorithm
The proposed algorithm executes through four main phases which include parameter setting, registration, authentication and key agreement.

### 3.1 Parameter Setting and Registration

During the parameter setting phase, the registration authority (RA) chooses $\SN_{\text{in}}$ and $\GN_{\text{in}}$ as unique sensor node (SN) and gateway node (GWN) identities respectively. Table 1 shows the network architecture for the proposed authentication algorithm.

As shown in Figure 1, the SNs can freely exchange packets with each other, which are then forwarded to the gateway node for transmission to remote users. Since the communication is over the public internet, the exchanged messages need to be sufficiently protected from any feasible security and privacy violations over these networks. At the onset of the proposed algorithm, registration of the users’ mobile devices through which they interact with SNs need registration at the RA. Similarly, the GWN is registered at the RA before being deployed to forward packets between remote users and SNs. Table 1 presents some of the symbols used in this paper and their particulars.

| Table 1. Notations |
|-------------------|
| Symbol | Description |
| $h(\cdot)$ | Hashing operation |
| RA | Registration authority |
| $\RA_{\text{sk}}$ | RA’s secret key |
| $\MD_{\text{ID}}$ | Mobile device identity |
| $\omega, \omega'$ | MD’s secret and public keys respectively |
| $N$ | Random numbers |
| $T$ | Timestamps |
| $\omega$ | RA and GWN shared secret key |
| $C$ | MD and GWN shared secret key |
| $E_1, \ldots, E_s$ | Message verification codes |
| $\tilde{A}$ | Session key |
| $\varrho$ | User’s secret key |
| $E_{\text{SK}}, E_{\omega}, E_{\varrho}, E_C$ | Encryption with keys $\text{SK}, \omega, \varrho, C$ respectively |
| $D_{\text{SK}}, D_{\omega}, D_{\varrho}, D_C$ | Decryption with keys $\text{SK}, \omega, \varrho, C$ respectively |
| $\|$ | Concatenation operation |
| $\oplus$ | XOR operation |

Then, it accepts user’s secret key $\varrho$ before computing parameter $U(\varrho)$ and MD’s pseudo-identity $\tilde{y}$ (step 4). Next, some of the computed parameters $\{\MD, \tilde{y}\}$ are sent to RA. Upon receipt of these parameters, the RA computes intermediate security parameters $p, q, r$ and $s$ for later authentication (step 5). Finally, RA sends $\{q, r, s, h(\cdot)\}$ to the MD.

### 3.2 Authentication and Key Agreement

Whenever the user seeks some sensor services or information, proper authentication is executed before this access is granted. After successful authentication, the sensor and user’s MD must agree on some session key to protect the exchanged data, as shown in Algorithm 2. The process begins by having the user set some expiration time $\Delta T$ for the exchanged messages. This is followed by user’s entry of secret key $\varrho$ into the MD which then derives parameters in step 1 before validating parameter $s$ in step 2. Next, random number $N_2$ is generated followed by security values in step 3. Afterwards, computed parameters $\{q, \tilde{g}, L_1, L_2\}$ are sent to the RA.
Algorithm 2: Authentication and Key agreement

BEGIN:
1) Set \( \Delta T \) & derive \( RP_q = \alpha_m, y_{sn} = \beta(t_{user}), \beta = \beta(t_{user}) \).
2) IF \( f < T \), THEN abort session.
3) ELSE: Generate \( N_1 \), & derive \( \gamma = \gamma_{(MD)} \).
4) MD: \( \alpha = \alpha_{(MD)} \).
5) Derive \( \Delta T = \Delta T_{(MD)} \).
6) Determine \( T \), & compute \( T = T_1, T_2 \).
7) ELSE: IF \( f > \Delta T \), THEN abort session.
8) IF \( C_3 \neq \gamma \), THEN abort session.
9) ELSE: trust MD.
10) Generate \( N_1 \) & derive \( \gamma = \gamma_{(MD)} \).
11) Compute \( \Delta T = \Delta T_{(MD)} \).
12) Determine \( T \), & compute \( T = T_1, T_2 \).
13) ELSE: IF \( f > \Delta T \), THEN abort session.
14) ELSE: generate \( N_1 \) & compute \( \Delta T = \Delta T_{(MD)} \).
15) Derive \( \Delta T = \Delta T_{(MD)} \).
16) Determine \( T \), & compute \( T = T_1, T_2 \).
17) IF \( f > \Delta T \), THEN abort session.
18) ELSE: generate \( \Delta T = \Delta T_{(MD)} \).
19) Compute \( \Delta T = \Delta T_{(MD)} \).
20) ELSE: IF \( f > \Delta T \), THEN abort session.
21) ELSE: Compress \( \Delta T = \Delta T_{(MD)} \).
22) Compute \( \Delta T = \Delta T_{(MD)} \).
23) ELSE: Trust MD.
24) Compute \( \Delta T = \Delta T_{(MD)} \).
25) Determine \( T \), & compute \( T = T_1, T_2 \).
26) IF \( f > \Delta T \), THEN abort session.
27) ELSE: Compress \( \Delta T = \Delta T_{(MD)} \).
28) Compute \( \Delta T = \Delta T_{(MD)} \).
29) ELSE: Trust MD.
30) ENDIF; ENDIF; ENDIF; ENDIF; ENDIF; ENDIF; ENDIF; ENDIF;

END

4. Results and Discussion

Upon receiving these values, RA re-computes \( MD_{SN} \) before computing the security parameter in step 4. However, in step 5, the current timestamp \( T_2 \) is determined upon which elapsed time \( T \) is computed and validated in step 6. If the validation is successful, RA derives and validates message verification code \( M_T \) in step 7 and 8 respectively. Provided this authentication is successful, RA and MD trust each other (step 9).

The next step is the commencement of RA and GWN authentication which begins by having RA derives random number \( N_1 \) followed by derivation of parameters in step 10. Next, message \( \{L_2, L_3, L_4 \} \) is sent to the GWN, upon which it calculates security parameters in step 11. Next, elapsed time \( T \) is computed (step 12) before being validated together with verification message \( L_4 \) in step 13. Afterwards, GWN generates random number \( N_1 \) followed by computation of message verification codes \( L_3 \) and \( L_6 \) in step 14. Thereafter, parameters \( \{L_2, L_3, L_6 \} \) are sent to the SN. Upon receipt of these values, the SN computes parameters in step 15 before computing elapsed time and validating the same together with \( L_6 \) in step 17. If this authentication is successful, SN generates random number \( N_6 \) before deriving parameters in step 18, a subset of which \( \{L_7, L_8, T_1\} \) is sent to the GWN. Here, the elapsed time is calculated (step 19) before being validated in step 20. If the received timestamp passes the freshness test, GWN re-computes random number \( N_4 \) before computing parameters in step 21. Next, message verification code \( L_6 \) is validated in step 22 such that if it is legitimate, GWN derives message verification code \( L_6 \) before sending \( \{L_3, L_4 \} \) to the MD.

Upon receipt of this message, the MD derives parameters in step 24, before determining and validating the freshness of the received message in step 25 and 26 respectively. Provided the message passes the freshness test, the MD computes session key \( \hat{A}_S \) together with message verification code \( L_6 \) (step 27). In step 28, this verification code is authenticated such that if it is valid, then the GWN and SN can trust each other.

4.1 Security Analysis

In this part, it is shown that the proposed algorithm is robust against legacy WSN privacy and security attack models. In addition, it is shown that the proposed algorithm offers forward key secrecy.

Forward key secrecy: in the proposed protocol, all the communicating entities share session key \( \hat{A}_S = h(N_S || N_S || N_S || L_3) \) for the protection of the exchanged traffic. It is clear that the computation of \( \hat{A}_S \)
The proposed algorithm deploys timestamps to \( T \) to check the freshness of all received messages. Suppose that an adversary has captured the current \( \{q, \tilde{g}, L_1, L_2\} \) sent from the MD towards the RA. The aim will then be to resend this message during subsequent authentication session. However, the RA has to decrypt \( L_1 \) (step 4) to obtain its timestamp that is then verified in step 6. As such, any replayed message will fail the freshness checks and the authentication process will be aborted. Similar freshness checks are executed on \( L_3 \) and \( L_5 \) and hence the proposed algorithm is robust against these attacks. Table 2 gives the security comparisons of the proposed algorithm with its peers.

### Table 2. Security features comparisons

| Security feature              | [17] | [12] | [16] | Proposed |
|------------------------------|------|------|------|----------|
| Forward key secrecy          | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Key agreement                | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Impersonation                | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Side-channel                 | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Traceability                 | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Offline guessing             | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Mutual authentication        | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Replay                       | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |

It is clear from Table 2 that the proposed algorithm offers many admirable WSN security features as compared with other related schemes. This was followed by the algorithm in [16], while the schemes in [12] and [17] had the worst security performance because of missing three crucial security features in each.

### 4.2 Performance Evaluation

In this sub-section, the computation and the communication overheads of the proposed algorithm are derived. This is then followed by the comparison of the obtained values with those of other related schemes.

#### Computation overheads
The proposed algorithm executed hashing \( T_5 \), symmetric key encryption and symmetric key decryption \( T_{\text{sm}} \) operations. Based on Algorithm 2, the MD executes \( 6T_h \) and \( 2T_{\text{sm}} \) operations while the RA executes \( 5T_h \) and \( 2T_{\text{sm}} \) operations. On the other hand, the GWN carries out \( 8T_h \) and \( 3T_{\text{sm}} \) operations while the SN computes \( 5T_h \) and \( T_{\text{sm}} \) operations. Consequently, the total computational overhead in the proposed algorithm is \( 24T_h \) and \( 8T_{\text{sm}} \) operations. Using the values in [17], a single \( T_h \) operation takes 0.0005 ms while a single \( T_{\text{sm}} \) operation takes 0.1303 ms. As such, the total computation overhead is 1.05 ms as shown in Table 3.

---

**Replay attacks:** to curb this attack, the proposed algorithm deploys timestamps to \( T \) to check the freshness of all received messages. Suppose that an adversary has captured the current \( \{q, \tilde{g}, L_1, L_2\} \) sent from the MD towards the RA. The aim will then be to resend this message during subsequent authentication session. However, the RA has to decrypt \( L_1 \) (step 4) to obtain its timestamp that is then verified in step 6. As such, any replayed message will fail the freshness checks and the authentication process will be aborted. Similar freshness checks are executed on \( L_3 \) and \( L_5 \) and hence the proposed algorithm is robust against these attacks. Table 2 gives the security comparisons of the proposed algorithm with its peers.

### Table 2. Security features comparisons

| Security feature              | [17] | [12] | [16] | Proposed |
|------------------------------|------|------|------|----------|
| Forward key secrecy          | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Key agreement                | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Impersonation                | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Side-channel                 | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Traceability                 | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Offline guessing             | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Mutual authentication        | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Replay                       | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |

It is clear from Table 2 that the proposed algorithm offers many admirable WSN security features as compared with other related schemes. This was followed by the algorithm in [16], while the schemes in [12] and [17] had the worst security performance because of missing three crucial security features in each.

### 4.2 Performance Evaluation

In this sub-section, the computation and the communication overheads of the proposed algorithm are derived. This is then followed by the comparison of the obtained values with those of other related schemes.

#### Computation overheads
The proposed algorithm executed hashing \( T_5 \), symmetric key encryption and symmetric key decryption \( T_{\text{sm}} \) operations. Based on Algorithm 2, the MD executes \( 6T_h \) and \( 2T_{\text{sm}} \) operations while the RA executes \( 5T_h \) and \( 2T_{\text{sm}} \) operations. On the other hand, the GWN carries out \( 8T_h \) and \( 3T_{\text{sm}} \) operations while the SN computes \( 5T_h \) and \( T_{\text{sm}} \) operations. Consequently, the total computational overhead in the proposed algorithm is \( 24T_h \) and \( 8T_{\text{sm}} \) operations. Using the values in [17], a single \( T_h \) operation takes 0.0005 ms while a single \( T_{\text{sm}} \) operation takes 0.1303 ms. As such, the total computation overhead is 1.05 ms as shown in Table 3.

---

**Replay attacks:** to curb this attack, the proposed algorithm deploys timestamps to \( T \) to check the freshness of all received messages. Suppose that an adversary has captured the current \( \{q, \tilde{g}, L_1, L_2\} \) sent from the MD towards the RA. The aim will then be to resend this message during subsequent authentication session. However, the RA has to decrypt \( L_1 \) (step 4) to obtain its timestamp that is then verified in step 6. As such, any replayed message will fail the freshness checks and the authentication process will be aborted. Similar freshness checks are executed on \( L_3 \) and \( L_5 \) and hence the proposed algorithm is robust against these attacks. Table 2 gives the security comparisons of the proposed algorithm with its peers.

### Table 2. Security features comparisons

| Security feature              | [17] | [12] | [16] | Proposed |
|------------------------------|------|------|------|----------|
| Forward key secrecy          | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Key agreement                | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Impersonation                | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Side-channel                 | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Traceability                 | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Offline guessing             | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Mutual authentication        | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Replay                       | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |

It is clear from Table 2 that the proposed algorithm offers many admirable WSN security features as compared with other related schemes. This was followed by the algorithm in [16], while the schemes in [12] and [17] had the worst security performance because of missing three crucial security features in each.
### Table 3. Computation Overheads

| Algorithm | Computation overheads (ms) |
|-----------|----------------------------|
| [12]      | 2.57                       |
| [16]      | 1.28                       |
| [17]      | 1.28                       |
| Proposed  | 1.04                       |

On the other hand, the schemes in [17], [12] and [16] take 1.28 ms, 2.57 ms and 1.28 ms respectively. Based on Figure 2, the scheme in [12] had the highest computation costs followed by the algorithms in both [16] and [17].

### Figure 2. Computations Overheads

As such, the proposed algorithm had the lowest computation overheads among its peers. This means that the proposed algorithm is applicable in battery powered sensor nodes.

**Communication overheads:** for this evaluation, the values in [17] are used in which timestamps, one-way hashing output, random numbers secret keys, identities and random numbers are all 128 bits wide. On the other hand, each ECC point multiplication is 160 bits wide. Based on Algorithm 2, messages \{q, ĝ, \(L_1, L_2\)\}, \{\(L_2, L_3, L_4\)\}, \{\(L_5, L_6, \hat{L}_5\)\} and \{\(L_7, L_8, \hat{T}_7\)\} are exchanged during the authentication and key agreement phase. Table 4 presents the communication overheads computations in the proposed algorithm.

### Table 4. Communication Overhead Derivation

| Message         | size (bits) |
|-----------------|-------------|
| MD→RA: \{q, ĝ, \(L_1, L_2\)\} | 512         |
| RA→GWN: \{\(L_2, L_3, L_4\)\} | 384         |
| GWN→SN: \{\(L_5, L_6, \hat{L}_5\)\} | 384         |
| SN→GWN: \{\(L_7, L_8, \hat{T}_7\)\} | 384         |
| GWN→MD: \{\(L_9\)\} | 256         |
| Total           | 1920        |

On the other hand, Table 5 shows that the algorithms in [17], [12] and [16] require 1856 bits, 3072 bits and 1856 bits respectively.

### Table 5. Communication Overheads Comparisons

| Algorithm | Communication overheads (bits) |
|-----------|-------------------------------|
| [12]      | 3072                          |
| [16]      | 1856                          |
| [17]      | 1856                          |
| Proposed  | 1920                          |

As shown in Figure 3, the schemes in [17] and [16] had slightly lower communication overheads compared with the proposed algorithm.

### Figure 3. Communication Overheads

Although the schemes in [17] and [16] had a better performance in terms of communication overheads compared with the proposed algorithm, their designs do not consider forward key secrecy, offline guessing and side-channel attacks. As such, in overall, the proposed algorithm offered strong security and relatively lower computation and communication overheads.

### 5. Conclusions

Wireless sensor networks have been heavily deployed in applications such as healthcare, military surveillance and environmental monitoring. Clearly, the information exchanged in these networks is sensitive and hence should not be accessed by authorized entities. However, since the transmission of this data is over the public internet, numerous security and privacy violations can be launched against the exchanged messages. Many schemes have been presented in literature to curb these attacks. However, it has been shown that these algorithms cannot offer all salient security features needed in this environment. To fill the gaps in most of these schemes, a wireless sensor network authentication algorithm has been developed in this paper. Its security evaluation has shown its superiority to other related algorithms in terms of resilience against side-channel, traceability, offline password guessing, replay and impersonations attacks. It also displayed average best performance with regard to computation overheads, and average performance in terms of communication...
overheads. Future work lies in the evaluation of this algorithm using security and performance metrics that were not within the subject scope of this work.

References

[1] J. Mo, and H. Chen, “A lightweight secure user authentication and key agreement protocol for wireless sensor networks,” Security and Communication Networks, 1-18, 2019.

[2] F. Wu, X. Li, L. Xu, L.P. Vijayakumar, and N. Kumar, “A novel three-factor authentication protocol for wireless sensor networks with IoT notion,” IEEE Systems Journal, 15(1), 1120-1129, 2020.

[3] B. Rashid and M.H. Rehmani, “Applications of wireless sensor networks for urban areas: A survey,” J. Netw. Comput. Appl., vol. 60, pp. 192-219, 2016.

[4] V.O. Nyangaresi, A.J. Rodrigues, and N.K. Taha, “Mutual Authentication Protocol for Secure VANET Data Exchanges,” in International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures, Springer, Cham, pp. 58-76, 2021.

[5] C. Miranda, G. Kaddoum, E. Bou-Harb, S. Garg, and K. Kaur, “A collaborative security framework for software-defined wireless sensor networks,” IEEE Transactions on Information Forensics and Security, 15, 2602-2615, 2020.

[6] V.O. Nyangaresi, and Z. Mohammad, “Privacy Preservation Protocol for Smart Grid Networks,” in 2021 International Telecommunications Conference (ITC-Egypt), IEEE, pp. 1-4, 2021.

[7] H.I. Kobo, A.M. Abu-Mahfouz, and G.P. Hancke, “A survey on software-defined wireless sensor networks: Challenges and design requirements,” IEEE Access, vol. 5, pp. 1872-1899, 2017.

[8] A. De Gante, M. Aslan, and A. Matrawy, “Smart wireless sensor network management based on software-defined networking,” in Proc. IEEE Commun. Biennial Symp., 71-75, 2014.

[9] V.O. Nyangaresi, and N. Petrovic, “Efficient PUF Based Authentication Protocol for Internet of Drones,” in 2021 International Telecommunications Conference (ITC-Egypt), IEEE, pp. 1-4, 2021.

[10] R. Murugesan, M. Saravanan, and M. Vijayaraj, “A node authentication clustering based security for adhoc network,” in Proc. IEEE Int. Conf. Commun. Signal Process, pp. 1168-1172, 2014.

[11] C. Zhu, V.C. Leung, L. T. Yang, and L. Shu, “Collaborative location based sleep scheduling for wireless sensor networks integrated with mobile cloud computing,” IEEE Trans. Comput., 64(7), 1844-1856, 2015.

[12] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, D. Won, “Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography,” Sensors, 14, 10081-10106, 2014.

[13] C. Wang, G. Xu, and J. Sun, “An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks,” Sensors, 17(12), 2946, 2017.

[14] M. Turkanović, B. Brumen, and M. Hölbl, “A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion,” Ad Hoc Netw., 20, pp. 96-112, 2014.

[15] Y. Lu, G. Xu, L. Li, and Y. Yang, “Anonymous three-factor authenticated key agreement for wireless sensor networks,” Wireless Networks, 25(4), 1461-1475, 2019.

[16] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, Y. Yang, “An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks,” Journal of Network and Computer Applications, 76, 37-48, 2016.

[17] X. Li, J. Niu, S. Kumari, F. Wu, A.K. Sangaiah, and K.K. Choo, “A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments,” Journal of Network and Computer Applications, 103, 194-204, 2018.

[18] Q. Jiang, J. Ma, X. Lu, and Y. Tian, “An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks,” Peer-to-Peer Networking and Applications, 8(6), 1070-1081, 2015.

[19] A.K. Das, “A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks,” Wireless Personal Communications, 82(3), 1377-1404, 2015.

[20] F. Wu, L. Xu, S. Kumari, and X. Li, “An improved and provably secure three-factor user authentication scheme for wireless sensor networks,” Peer-to-Peer Networking and Applications, 11(1), 1-20, 2018.

[21] X. Li, J. Peng, M.S. Obaidat, F. Wu, M.K. Khan, and C. Chen, “A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems,” IEEE Systems Journal, 14(1), 39-50, 2019.

[22] D. Wang, D. He, P. Wang, and C.-H. Chu, “Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment,” IEEE Trans. Depend. Sec. Comput., 12(4), 428-442, 2015.

[23] G. Jaswal, A. Kaul, and R. Nath, “Multimodal bio-
metric authentication system using hand shape, palm print, and hand geometry,” in Computational Intelligence: Theories, Applications and Future Directions, Springer, Singapore, 557-570, 2019.

[24] D. Jagadiswary and D. Saraswady, “Biometric authentication using fused multimodal biometric,” Procedia Comput. Sci., 85, pp. 109-116, 2016.

[25] F. Li and P. Xiong, “Practical secure communication for integrating wireless sensor networks into the internet of things,” IEEE Sensors Journal, 13(10), 3677-3684, 2013.

[26] C.L. Chen, T.F. Shih, Y.T. Tsai, and D.K. Li, “A bilinear pairing-based dynamic key management and authentication for wireless sensor networks,” Journal of Sensors, 1-15, 2015.

[27] S. Ramachandran and V. Shanmugam, “A two way authentication using bilinear mapping function for wireless sensor networks,” Computers & Electrical Engineering, 59, pp. 242-249, 2017.

[28] V.O. Nyangaresi, A.J. Rodrigues, and S.O. Abeka, “Neuro-Fuzzy Based Handover Authentication Protocol for Ultra Dense 5G Networks,” in 2020 2nd Global Power, Energy and Communication Conference (GPECOM), IEEE, 339-344, 2020.

[29] A.K. Das, “A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor,” International Journal of Communication Systems, 30(1), e2933, 2015.

[30] M.S. Farash, M. Turkanović, S. Kumari, and M. Holbl, “An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of things environment,” Ad Hoc Networks, 36, 152-176, 2016.

[31] R. Amin, S.H. Islam, G.P. Biswas, M.K. Khan, L. Leng, and N. Kumar, “Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks,” Computer Networks, 101, 42-62, 2016.

[32] M. Amjad, H.K. Qureshi, M. Lestas, S. Mumtaz, and J.J. Rodrigues, “Energy prediction based MAC layer optimization for harvesting enabled WSNs in smart cities,” in Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring), pp. 1-6, 2018.

[33] Q. Jiang, S. Zeadally, J. Ma, and D. He, “Lightweight three factor authentication and key agreement protocol for internet-integrated wireless sensor networks,” IEEE Access, 5, pp. 3376-3392, 2017.

[34] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, 5, 21954-21961, 2017.

[35] M.A. Ambusaidi, X. He, P. Nanda, and Z. Tan, “Building an intrusion detection system using a filter-based feature selection algorithm,” IEEE Trans. Comput., 65(10), 2986-2998, 2016.

[36] T. Ma, Y. Yu, F. Wang, Q. Zhang, and X. Chen, “A hybrid methodologies for intrusion detection based deep neural network with support vector machine and clustering technique,” Sensors, 6(10), 1701, 2016.

[37] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, “Long short term memory recurrent neural network classifier for intrusion detection,” in Proc. Int. Conf. Platform Technol. Service (PlatCon), IEEE, pp. 1-5, 2016.