SLAE6: Secure and Lightweight Authenticated Encryption Scheme for 6LoWPAN Networks

Fatma Foad Ashrif[1,2][a], Elankovan A. Sundarajan[1][b], Rami Ahmed[3][c]
and Mohammad Kamrul Hasan[1][d]
[1]Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
[2]Department of Computer Science, Sebha Universit, Ubari, Libya
[3]College of Computer Information Technology, American University in the Emirates, 503000, Dubai, U.A.E.

Keywords: Internet of Things, Authenticated Encryption, Key Establishment, 6LoWPAN, Wireless Sensor Networks.

Abstract: The emergence of the Internet of things is highly related to the development of wireless sensor networks (WSNs) and their evolving protocols, such as Internet Protocol version 6 (IPv6) over Low-Power Wireless Personal Area Networks (6LoWPAN). Providing security within a sensor network, including achieving authentication between WSN nodes, is critical. The node and the server create an encryption session key for future communications. Therefore, developing a lightweight and efficient authentication and key establishment (AKE) scheme is imperative. Symmetric cryptographic and public key-based AKE methods have been developed to address these issues. Nevertheless, some known attacks and large communication and computational overheads remain as problems for the developed solutions. This study proposes a secure and lightweight authenticated encryption scheme for 6LoWPAN (SLAE6) that uses a lightweight hash function and an authenticated encryption primitive, known as ACE, to enable the AKE process to occur securely. SLAE6 is effective in dealing with computing and communication complexities while simultaneously withstanding well-known attacks. First, SLAE6 validates the authenticity of information from sensor networks (SNs) and then establishes a secret key between an SN and the server to guarantee security. The proposed system is proven reliable on the basis of the Canetti–Krawczyk and Dolev–Yao threat models. In addition, SLAE6 is logically demonstrated to be exact through Burrows–Abadi–Needham logic. Compared with other schemes, SLAE6 is lightweight, efficient, and requires less bandwidth and shorter execution time.

1 INTRODUCTION

IPv6 addresses used by sensor nodes allow them to transmit sensing information to other devices or to a central location across the Internet through IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) (Bagwari et al., 2022), which is meant to support all the functions of IPv6 over LoWPAN, including packet fragmentation, reassembly, and encapsulation (Tanaka et al., 2019). 6LoWPAN applications must provide privacy and security because they transmit information via the Internet. However, no security or privacy feature is built into the basic 6LoWPAN design to prevent unauthorized entities from obtaining information or unauthorized users from gaining access to network resources (Tanveer et al., 2020). Considering the resource constraints and insufficiently organized network architecture in 6LoWPAN, securing these networks has become more challenging (Monika et al., 2022). A 6LoWPAN designed for WSNs and IoT should have inexpensive sensor nodes that require relatively low power, resulting in low computation performance (Wazirali et al., 2021). Moreover, sensitive areas occasionally require nodes that cannot be regularly powered, and thus, conserving energy while maintaining security is essential (Ahmad et al., 2021)(Wazirali & Ahmad, 2022).

[a] https://orcid.org/0000-0001-7711-7520
[b] https://orcid.org/0000-0003-2711-0659
[c] https://orcid.org/0000-0003-3913-6397
[d] https://orcid.org/0000-0001-5511-0205

Ashrif, F., Sundarajan, E., Ahmed, R. and Hasan, M.
SLAE6: Secure and Lightweight Authenticated Encryption Scheme for 6LoWPAN Networks.
DOI: 10.5220/0011632200003399
In *Proceedings of the 12th International Conference on Sensor Networks (SENSORNETS 2023)*, pages 67-78
ISBN: 978-989-758-635-4; ISSN: 2184-4380
Copyright © 2023 by SCITEPRESS – Science and Technology Publications, Lda. Under CC license (CC BY-NC-ND 4.0)

Furthermore, a secure connection layer requires powerful hardware and consumes considerable power (Ahmad et al., 2021). Lightweight cryptography (LWC) algorithms offer an efficient means of reducing computation complexity and preserving security. An authentication scheme that is lightweight and secure is an effective way to establish scalable and reliable communication between IoT devices (Amanlou et al., 2021). 6LoWPAN must ensure authenticity, data integrity, freshness, availability, and confidentiality. Confidentiality ensures that data are transmitted securely between authorized WSNs and servers. In 6LoWPAN, authentication and key establishment (AKE) is a mechanism for identifying a network's security; implementing a lightweight AKE mechanism is imperative (Tanveer et al., 2021). AKE is crucial for achieving reliable and secure communication in the IoT or WSN. Given the computational complexity of SNs, conventional AKE schemes, such as blockchains and passwords, are unsuitable for 6LoWPAN devices. Compared with public-key cryptography, less energy and computational resources are consumed by symmetric-key cryptography (Hasan et al., 2021). An authenticated encryption with associative data (AEAD) scheme is presented using a lightweight cryptography primitive known as ACE (Aagaard et al., 2019). With an LWC-based authenticated encryption (AE) scheme, data encryption and authentication are performed simultaneously. Therefore, we propose a secure and lightweight AE scheme for 6LoWPAN (SLAE6) that is secure and efficient by using AEAD mechanisms. SLAE6 ensures anonymity, untraceability, and end-to-end security from the SNs to the server. The following contributions are highlighted in the current study.

- An AKE scheme that provides end-to-end security by using LWC-based lightweight AEAD mechanisms, XOR operation, and hash function (HF) is proposed.
- An informal analysis is conducted to demonstrate the robustness of SLAE6 under the threat models of Dolev–Yao (DY) and Canetti–Krawczyk (CK). A formal analysis based on Burrows–Abadi–Needham (BAN) logic confirms MA.
- SLAE6 has less computation cost, bandwidth requirements and execution time than other related schemes.

The remainder of this paper is organized as follows. Section 2 provides a presentation related to the AKE scheme. Section 3 describes the adopted system model. Section 4 explains our proposed scheme in detail. Section 5 provides a security analysis of the proposed SLAE6 and its comparison with related schemes. Section 6 concludes the paper. Table 1 summarizes the abbreviations used throughout this paper.

Table 1: Abbreviation table.

Acronyms	Paraphrase
WSN	Wireless Sensor Network
IoT	Internet of Things
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
LWC	Lightweight Cryptography
AKE	Authentication and Key Establishment
AEAD	Authenticated Encryption with Associative Data
MITM	Man in the Middle
HF	Hash Function
DoS	Denial of Service
ECC	Elliptic-curve Cryptography
PKI	Public Key Infrastructure
PUF	Physical Unclonable Function
DY	Dolev–Yao model
CK	Canetti–Krawczyk model

2 RELATED WORK

This section provides an overview of existing IoT AKE schemes. Various security specifications for 6LoWPAN were discussed in Hennebert and Santos (2014). The authors proposed a secure AKE scheme (SAKES) by using public-key cryptography (Hussen et al., 2013); however, this scheme is computationally costly for devices with limited storage. Qiu and Ma (2016) proposed an AKE scheme based on a hybrid cryptography approach for 6LoWPAN that is insecure against chosen plaintext, sinkhole, and node capture attacks in accordance with Gao et al. (2020). Chom Thungon et al. (2020) claimed that this scheme cannot resist replay and man-in-the-middle (MITM) attacks. However, although Gao et al. (2020) addressed these security concerns, both schemes still require considerable computing and communications. In Chom Thungon et al. (2020), an authentication scheme that utilizes lightweight keys and is optimized for 6LoWPAN was proposed to authenticate resource-constrained sensor devices that use HF and XOR operations. However, their scheme cannot detect some attacks, such as denial-of-service (DoS) attacks. In Tanveer et al. (2020), the authors presented an AKE scheme that uses HF, XOR operations, and ASCON for 6LoWPAN. However, this scheme lacks untraceability features. In Chom Thungon et al.

(2020) and Tanveer et al. (2020), BAN logic was used to verify the security of the proposed scheme. Alshrif et al. (2021) developed an AKE framework based on a symmetric algorithm, HF, and XOR operations; then, they proposed a lightweight encryption technology scheme that enables energy efficiency and secures WSN communication in 6LoWPAN. Ahmed et al. (Ahmad et al., 2022) presented a secure and cost-saving framework for low-cost 6LoWPAN based on adaptive trust. Anonymity and untraceability are not considered in many of these protocols. Performing a public key infrastructure (PKI)-based technique is computationally intensive. Most elliptic-curve cryptography (ECC)-based schemes utilize time-consuming bilinear pairing operations. Nevertheless, ECC-based authentication and key agreement cost lower than PKI-based schemes. Even hardware-based solutions, such as physical unclonable function (PUF) (Fragkos et al., 2022), cannot be implemented in some environments, such as underwater or climate monitoring. Implementing robust security and privacy in a WSN or IoT environment at lower communication and computational costs remains challenging. In consideration of this issue, we propose a 6LoWPAN-compatible AKE scheme that uses an efficient and secure AEAD called ACE, ephemerals, pseudo identities, XOR, and hashing. This scheme requires minimal execution time and bandwidth.

3 SYSTEM MODEL

The proposed scheme and network of SLAE6 and the threat models are presented in this section.

3.1 Network Model

The SLAE6 network model is presented in Figure 1 to illustrate mutual AKE in 6LoWPAN. The proposed protocol includes 6LoWPAN sensor nodes (*6LNs*), 6LoWPAN router (*6LR*), 6LoWPAN border router (*6LBR*), and 6LoWPAN server (*6LS*). As shown in Figure 1, the *6LNs* gather data, while the *6LR* aggregates the sensor data before forwarding them to the *6LBR*, which delivers them to the *6LS*. In addition, the *6LR* facilitates Internet connectivity within domains via the *6LNs*. The *6LR* provides IPv6 cloud interconnectivity with the *6LS*. Communications among the *6LR*, *6LBR*, and *6LS* are assumed as secure. In addition, the *6LNs*, *6LR*, and *6LBR* are assumed to reach the *6LS*. In addition to registering with the *6LS* via a secure channel, the *6LR* communicates with the *6LNs* via the neighbor discovery (ND) protocol to exchange temporary identities. The *6LR* also registers itself with the *6LBR*. All 6LoWPAN devices access the *6LS* global routing prefix through the *6LBR*. In addition, the *6LNs* generate IPv6 addresses by using IEEE extended unique identifiers referred to as PAN IDs (Hui & Thubert, 2011).

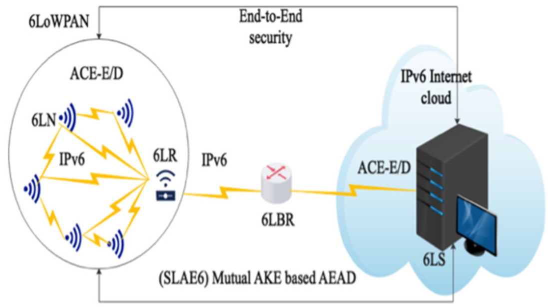

Figure 1: Network model of SLAE6 provides AKE-based AEAD from the *6LNs* to the *6LS*, ensuring end-to-end security in 6LoWPAN networks.

3.2 Adversary Model

In this subsection, we will assume a model of attacks that may expose communication between both ends of the devices. Thus, securing communications between the *6LNs* and *6LS* is imperative. Hence, the *6LS* must also be prevented from receiving data from illegitimate *6LNs*. To ensure the credibility of the *6LNs*, we devise an AKE procedure that provides a secret *SK* that can be used for future communication after validating the authenticity of the *6LNs*. The procedure design of SLAE6 falls under the following definitions.

Definition 1: As a result of the one-time pad theorem, if a random value is XORed with a value, then the resulting value will also be random.

Definition 2: To have a secure HF, $h(.)$: (a) given an input message K, generating $h(K)$ of a fixed length given an input message K of an arbitrary length is possible; and (b) in the case of K, finding the value of $h(K') = h(K)$ is computationally impossible.

Definition 3: In the DY model (Dolev & Yao, 1983), adversary ($\mathcal{A}$) can (a) have valid credentials but be malicious; and (b) control the open communication medium, and thus, alter, intercept, insert, or erase messages sent over this medium.

Definition 4: In accordance with the CK model (Sarr et al., 2010), adversary ($\mathcal{A}$) can compromise session-specific state information and DY model capabilities. Moreover, secrets must not be disclosed by compromising the secrecy of another party if they compromise the security of a party. In addition to the adversary's capabilities under the DY and CK threat models, A can compromise the session state, secret key, and session key (*SK*) by using session hijacking. Therefore, short-term (temporal) and long-term (permanent) secrets must be considered in generating the *SK* between two entities.

3.3 ACE: An Authenticated Encryption Algorithm

To resolve the problems mentioned in Section 2, the AEAD algorithm is used. With AEAD, associated data (*AD*), such as routing data, can be verified in terms of validity and integrity. The AEAD scheme has elicited considerable interest in cryptography due to the National Institute of Standards and Technology (NIST)-funded Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) and the NIST-LWC competition for standardizing lightweight AEAD schemes. A report was published in October 2019 on the lightweight cryptography standardization process of NIST (Turan et al., 2021). In the current study, we use ACE, an authenticated encryption and hash algorithm. It is one of the algorithms selected from NIST's first round of lightweight cryptography competition. In ACE encryption and decryption algorithms, the input includes *AD*, key, and nonce, each of which is 128 bits. The output includes ciphertext, plaintext, and authentication tags, as shown in Figure 2. Therefore, dividing or truncating the output of SHA-256 into 128 bits is necessary to generate ACE's input parameters for encryption and decryption.

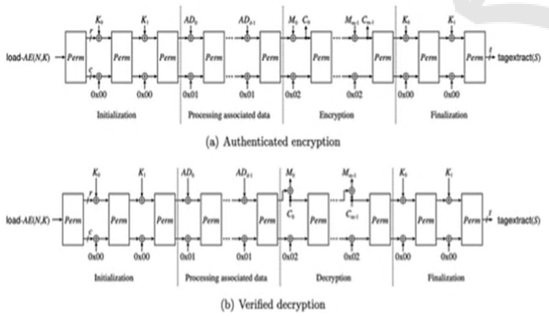

Figure 2: ACE architecture (Aagaard et al., 2019).

4 PROPOSED SCHEME

ACE is used as the encryption scheme by the *6LS* and *6LNs* after verifying the authenticity of the 6*LNs* in SLAE6. We generate a unique output string by combining SHA-256 with bitwise XOR operations and SLAE6 secret parameters. SLAE6 has two phases: registration and AKE for the static *6LNs*. The subsequent sections provide additional details about each phase. As indicated in Table 2, the current study uses the following phases.

4.1 Registration Phase

A registration phase is required before the *6LN* can be deployed. The state is loaded byte-wise with a 128-bit nonce $Ns = N_{S0} \,||\, N_{S1}$ and 128-bit key $Km = K_{M0} || K_{M1}$, and the remaining 8 bytes are set to zero. The *6LS* performs the following operations to register the *6LNs*.

Table 2: List of notations.

Notation	Description
6LN, 6LR, 6LBR, 6LS	6LoWPAN sensor node, 6LoWPAN router, 6LoWPAN border router, and 6LoWPAN server, respectively
TID_{6LN}, TID_{6LR}	Temporary identities of the *6LN* and *6LR*.
ID_{6LN}, ID_{6LS}	Secret real identities of the *6LN* and *6LS*.
O_{SP}, O_{SP1}	Secret authentication parameters.
(Tag_{6LN}, Tag'_{6LN}), (Tag_d, Tag_{6LS})	Tag generated authentication parameters for the *6LN* and *6LS*.
T1, T2, T3	Timestamps for *6LN*, *6LBR*, and *6LS*, respectively.
$T\Delta$, *Tr*	Maximum transmission delay limit and received time of messages.
(Si,Si'''),(Si',Si'')	Initialization states for the *6LN* and *6LS*.
K_{6LN}, *RN1, RN2, R1, R2*	The key and random numbers of the *6LN* are used in the AKE process.
MAC_{6LN},MAC_{6LS}	MAC addresses of the *6LN* and *6LS*.
D	Verification of the temporary identities of messages.
A, B	A and B are connected to represent the message's plain text (*PT*).
CT	Cipher text of the message.
D_{Si} (), E_{Si} ()	Decryption and encryption of *CT* by using the initialization state *Si*.
H(.), ||, ⊕	HF, concatenation, and bitwise XOR, respectively.

Picks up of the *6LS* real identity ID_{6LS}, and it is random number R_{6LS}, such that we can compute the master key (*Km*) by calculating $Km = H(ID_{6LS} || R_{6LS})$. *Km* is divided into 64 bits by the *6LS*, namely, K_{M0} and K_{M1}. $K_{6LS} = K_{M0} \oplus K_{M1}$ is computed, where K_{6LS} is a temporary key for the *6LS*. A nonce N_S of 128 is generated. The *6LS* divides N_S into two 64-bit chunks: N_{S0} and N_{S1}. A unique ID_{6LN}, K_{6LN}, is selected, and the temporary identity (TID_{6LN}) of 64 bits for the *6LN* is computed. $TID_{6LN} = ID_{6LN} \oplus K_{6LN} \oplus K_{6LS}$.

Subsequently, the *6LS* computes its temporal secret component by computing $T_{SP}=H(Km||K_{6LN}\ ||ID_{6LS})$ and calculating $O_{SP}= O_{SP1} \oplus O_{SP2} \oplus O_{SP3} \oplus O_{SP4}$, where O_{SP1}, O_{SP2}, O_{SP3}, and O_{SP4} are 64-bit chunks divided into four equal parts of O_{SP}. To load the state, a 128-bit nonce $Ns = N_{S0}||N_{S1}$, a 128-bit key $Km= K_{M0}||K_{M1}$, and the remaining 8 bytes are set to zero. Then, *Load_AE*(*Ns*||*Km*). Finally, the *6LS* stores *6LN*-related secret information, i.e., $\{ID_{6LS}, O_{SP}, K_{6LN}, K_{6LS}, MAC_{6LN}\}$ into its database and $\{ID_{6LS}, O_{SP}, K_{6LN}, TID_{6LN}, MAC_{6LS}\}$ in *6LN*'s memory by utilizing a secure channel. Secure channels are also used by the *6LS* to store TID_{6LN} in the *6LR* memory.

4.2 Initialization State

This phase is dedicated to initializing ACE's state *Si*, which is composed of 320 bits and called the initialization state *Si*. The *6LN*, which is a random number *R1* of 64 bits, is generated and computes $M_{6LN}=(R1||TID_{6LN})$, where M_{6LN} is an initialization for the *6LN*. The size of *Si* is 320 bits ($load-AE(Ns||Km)$ = 256 bits + M_{6LN} = 64 bits), which is used during the initialization phase as input to the encryption algorithm. Thereafter, the state is absorbed once the permutation ACE is applied to the two key blocks. The following are the initialization steps:

- $Si \leftarrow ACE\ (load\text{-}AE(Ns||Km))$,
- $Si \leftarrow ACE\ (K_{M0} \oplus M_{6LN})$,
- $Si \leftarrow ACE\ (K_{M1} \oplus M_{6LN})$.

4.3 Associative Data Generation

To generate *AD*, we perform the following operations:

The *6LN* computes $P_{HC}=H(R1||MAC_{6LN})$. Then, it divides P_{HC} into two, namely, P_{HC1} and P_{HC2}, with each containing 128 bits. The *6LN* calculates $AD = P_{HC1} \oplus P_{HC2}$. *AD* is 128 bits in size. To preserve the integrity of the associative data, the encryption algorithm uses *AD* as one of the input in the associative data processing phase.

4.4 AKE Phase

In this phase, the *6LN* enables anonymous AKE with the *6LS* by using the *6LR* and *6LBR* as intermediate nodes. The *6LN* and *6LS* can exchange data securely once a secret key is established. The authentication process involves four messages exchanged by SLAE6. The details about the messages exchanged in SLAE6 are provided as follows.

Step 1: The *6LN* generates 46 bits random number *RN1* and 32 bits timestamp *T1* for computing $A=ID_{6LN} \oplus RN1 \oplus O_{SP}$ and $B= ID_{6LN} \oplus RN1$, where *A* and *B* are 64 bits. The *6LN* and *6LS* use *Si* as input to the ACE encryption algorithm. *AD* is computed in the associative data generation processing. When (*A*||*B*) is processed at the plaintext level, it produces the ciphertext $(CT)=E_{Si}\{AD, (A||B)\}$ and Tag_{6LN}, (which is generated automatically by ACE. That tag is extracted from the same byte positions used when the key is loaded. Consequently, *CT* ensures that plaintext (*A*||*B*) is confidential. On the receiving end, Tag_{6LN} ensures the integrity and authenticity of the ciphertext *CT*. The Tag_{6LN} function is similar to the message authentication code (MAC). The *6LN* calculates $D=TID_{6LN} \oplus TID_{6LR}$, where TID_{6LR} is the temporary identity of the *6LR*. After these operations, the *6LN* constructs a message $Mgs_1{:}(T_{6LN}\ ||D\ ||\ (CT\ ||Tag_{6LN})||R1)$ that is sent to the *6LR* for further processing.

Step 2: Upon receiving Msg_1 from the *6LN*, the *6LR* extracts *D* and computes $TID_K = D \oplus TID_{6LN}$. The *6LR* compares TID_K with the stored TID_{6LR} in its memory. As long as TID_K and TID_{6LR} have the same contents, *6LR* adds its TID_{6LR} with the received Msg_1 to generate the new message $Msg_2{:}(TID_{6LR}\ ||\ Msg_1)$ and forwards it to the *6LBR*. Otherwise, the message is sent back to the *6LN* if the *6LR* aborts the AKE process.

Step 3: The *6LBR* receives Msg_2 from the *6LR* and checks the existence of TID_{6LR} in its database. The AKE process is aborted if the *6LBR* cannot find TID_{6LR} in the list, and the unverified TID_{6LR} is added to the block list. By contrast, upon successfully verifying the TID_{6LR} for Msg_2, the *6LBR* selects a time stamp *T2* and computes $S_{6LBR} = K_{6LBR} \oplus TID_{6LR}$, where TID_{6LBR} is the temporary identity of the *6LBR*, and K_{6LBR} is the pre-shared key between the *6LBR* and *6LS*. In the next step, the *6LBR* generates $Msg_3{:}(TID_{6LBR}||T2||Msg_2||S_{6LBR})$ and forwards it to the *6LS* to be processed further.

Step 4: Upon receiving Msg_3 from the *6LBR*, the *6LS* retrieves *6LBR*-related secret information by utilizing TID_{6LBR}. Moreover, the *6LS* checks *T2* validity by checking that Msg_3 is received within the allowance maximum transmission delay ($T\Delta$) by calculating $Tr-T2 \leq T\Delta$, where *Tr* represents the received timestamp for Msg_3. To verify the integrity of Msg_3, the *6LS* derives $S_{6LBR}' = K_{6LBR} \oplus TID_{6LR}$. If S_{6LBR}' and the received S_{6LBR} do not match, then the *6LBR* is added to the suspicious device list. After verifying the integrity of Msg_3, the *6LS* extracts Msg_2 from Msg_3 and checks its freshness by confirming whether $Tr-T1 \leq T\Delta$. the *6LS* rejects Msg_2 if the condition is not met. A valid TID_{6LR} is also checked in the current list of *6LR* devices by the *6LS*. If the verification of TID_{6LR} is successful, then the *6LS* extracts D from Msg_2, derives TID_{6LN} by computing $TID_{6LR} \oplus D$, and verifies if TID_{6LN} exists in its database. The $6LS$ retrieves ID_{6LN}, K_{6LS}, K_{6LN}, and O_{SP} information after verifying TID_{6LN} in its database. Using $R1$ and TID_{6LN}, the $6LS$ generates M_{6LN} from Msg_1. Moreover, the $6LS$ determines AD by using $R1$ from Msg_1 and the stored MAC_{6LN} in the *6LS's* database by computing $P_{HC}'=H(R1||MAC_{6LN})$. Hence, $AD'=P_{HC1}'\oplus P_{HC2}'$. In addition, the $6LS$ performs the decryption operation $Dsi'\{(AD', CT)\}$. As the decryption algorithm extracts the plaintext information, Tag_d is first generated. When ACE processes AD and the ciphertext, the authentication Tag_d is generated automatically. When Tag_{6LN} is received with Msg_1, the $6LS$ checks the condition $Tag_{6LN}=Tag_d$. Inverse-free authentication schemes generate the same authentication tag during encryption and decryption if AD and the ciphertext are not modified. Nonetheless, if the communicated message is modified, then the generated authentication tag differs, resulting in the proposed AKE failing to authenticate. A successful decryption reveals the plaintext information if the condition holds. Otherwise, the AKE process is aborted if this scenario occurs. When CT is decrypted, A and B are revealed as plaintext. The $6LS$ selects ID_{6LN} and computes the $ID_{6LN} \oplus B$ operation to determine $RN1$ for calculating $O_{SP}' =ID_{6LN} \oplus RN1 \oplus A$. In addition, the $6LS$ checks if $O_{SP}=O_{SP}'$ to ensure the legitimacy of the $6LN$. AKE is aborted if the condition does not hold. The $6LS$ registers the $6LN$ as a legitimate device. Upon verifying the validity of the $6LN$, the $6LS$ selects the 32-bit timestamp $T3$ and 64 bits random numbers $RN2$, $R2$, and RN. Then, $T''=H(Km|| RN ||ID_{6LS})$ is derived and a new security parameter O_{SP1} is calculated by computing $O_{SP1}=T''_1\oplus T''_2\oplus T''_3\oplus T''_4$. The $6LS$ calculates $B1=RN\oplus K_{6LS}$, $A1=B1\oplus RN1$, and $M_{6LS}'=H(R2||A1)$. To generate Si', the $6LS$ generates nonce Ns'' and calculates $Si''=load\text{-}AE(Ns||Km)$. Subsequently, the $6LS$ calculates AD by computing $P_{HC}'' =H(R1||MAC_{6LN})$, $AD1=P_{HC}''\oplus P_{HC}''$. To ensure future secure communication, the $6LS$ computes an SK by computing $SK=H(ID_{6LN}||B1||O_{SP1}||RN1||RN2)$. Furthermore, during the initialization phase, Si'' is considered by the encryption algorithm, AD'' during the AD processing phase, and $(O_{SP1}||RN2)$ while processing plaintext information to generate $\{CT1\}=E_{Si}''\{AD'',(O_{SP1}||RN2)\}$ and Tag_{6LS}. Moreover, the $6LS$ constructs the message $Msg_4{:}(T3||A1||(CT1||Tag_{6LS})||R2)$ and forwards it to the $6LBR$. Then, the $6LBR$ and $6LR$ forward Msg_4 to the $6LN$. Lastly, the $6LS$ saves the parameters $\{ID_{6LN}, O_{SP}, O_{SP1}, K_{6LS}\}$ in its memory.

Step 5: Upon obtaining Msg_4, the $6LN$ verifies the freshness of timestamp $T3$ by checking whether $Tr-T3 \le T\Delta$, where Tr represents the period in which Msg_4 has been received and $T\Delta$ represents the maximum allowed time. Significantly, the $6LN$ rejects Msg_4 if $T3$ exceeds the maximum time allowed. The $6LN$ obtains $R2$ and $A1$ from Msg_4 and computes $M_{6LS}''=H(R2||A1)$. M_{6LS} also calculates $B1=RN1\oplus A1$, $Ns'''=H(O_{SP}||TID_{6LR})$ and $Si'''=(load\text{-}AE(Ns||Km)'''||\ M_{6LS}''')$. Subsequently, the $6LN$ calculates AD by computing P_{HC}'' =H$(R1||MAC_{6LN})$, $AD1= P_{HC}''' \oplus P_{HC}'''$. As the input to the decryption algorithm, Si'' receives an associative data processing phase of Si, and $CT1$ receives a ciphertext processing phase of CT. The decryption operation is performed on $DSi'''\{AD''', CT1\}$ to generate Tag_{6LN}''. In the final step, the $6LN$ checks whether $Tag_{6LN}=Tag_{6LS}$. As long as the condition is met by decrypting the message, the plaintext is revealed, i.e., $(O_{SP1}||RN2)$. Then, the $6LN$ computes SK by computing $SK=H(ID_{6LN}||B1|| O_{SP1}||RN1||RN2)$ to secure future communications with the $6LS$. Finally, the $6LN$ stores the parameters $\{ID_{6LN}, O_{SP1}, TID_{6LN}, K_{6LN}\}$ in its memory.

5 COMPARATIVE AND SECURITY ANALYSES

This this section has two parts. The first provide security analysis, and the second provides performance evaluation.

5.1 Security Analysis

SLAE6 is analyzed in two phases in this section. The first phase describes SLAE6's capabilities and characteristics against malicious attacks. The second phase incorporates BAN logic to demonstrate the logical correctness of the SLAE6 scheme.

5.1.1 Informal Security Analysis

Throughout this subsection, the robustness of this protocol is demonstrated under all DY and CK assumptions, as discussed in the threat models. The DY and CK models assume that network communication occurs over unsecured channels, and none of the communicating entities can be trusted. On the basis of the adversarial properties found in Definitions 3 and 4, the proposed protocol was evaluated against replay, impersonation, MIMT, and DoS attacks, except for the ephemeral-secret-leakage (ESL) attack, which was evaluated only under CK. Furthermore, SLAE6 ensures MA, perfect forward secrecy, untraceability, and anonymity. The following theorems are used to achieve this objective:-

Theorem 1: *SLAE6 ensures MA*.
Proof: Each participant authenticates the other by using a public channel during the AKE phase. To confirm the authenticity of the *6LN*, the *6LS* checks if the device's *ID* is in its memory and confirms that $Tag_d=Tag_{6LN}$. To ensure the authenticity of the *6LS*, the *6LN* verifies the condition $Tag_{6LN}=Tag_{6LS}$. Consequently, the *6LS* and *6LN* reach *MA* with the help of *6LR* and *6LBR* in SLAE6.

Theorem 2: *SLAE6 ensures forward/backward secrecy* (*F/B S*).
Proof: An *SK* for an AKE session can be determined by computing the value of $SK=H(ID_{6LN}||B1||O_{SP1}||RN1||RN2)$ for every AKE session. New parameters, such as *B1*, O_{SP1}, *RN1*, and *RN*, are incorporated into the new AKE process. An adversary ($\mathcal{A}$) cannot compromise the future *SK* if the current *SK* is compromised. Consequently, adversaries cannot construct past or future *SKs* in our scheme.

Theorem 3: *SLAE6 ensures untraceability and anonymity.*
Proof: Consider the scenario in which messages have been eavesdropped by an adversary ($\mathcal{A}$), Msg_1 and Msg_2, where Msg_1:$(T1||D\ ||(CT||Tag_{6LN})||R1)$ and Msg_2:$(TID_{6LR}||(M_{sg1}))$. Furthermore, $D=TID_{6LN} \oplus TID_{6LR}$ and $(CT, Tag_{6LN})=E_{Si}\{AD,(A||B)\}$. By using captured messages exchanged over insecure channels, the attacker attempts to derive the *6LS's* ID_{6LS} and the *6LN's* ID_{6LN}. When transacting with authentication messages, the *6LN* uses temporary identity TID_{6LN}, calculated as $TID_{6LN}= ID_{6LN}\oplus K_{6LN}\oplus K_{6LS}$, where all the parameters are secret to the *6LS* and *6LN*. Therefore, $\mathcal{A}$ experiences difficulty generating TID_{6LN} without these parameters. In addition, $\mathcal{A}$ cannot distinguish messages from different participants by including the random number *R1* and timestamp *T1* in messages. The *6LN* in SLAE6 selects fresh random numbers in every new session. The *6LN* performs different computations to generate fresh random numbers, such as $A=ID_{6LN}\oplus RN1\oplus O_{SP}$ and $B=ID_{6LN}\oplus RN1$. $(A||B)$ is used in plaintext processing and produces ciphertext. Hence, for every new session, *Si*, *CT*, and Tag_{6LN} are different because fresh random numbers are used in SLAE6. Furthermore, a new *T1* of the *6LN* is created after a session is completed, increasing the untraceability of messages. Consequently, SLAE6 provides anonymity and untraceability.

Theorem 4: *SLAE6 is resistant to replay attacks.*
***Proof*:** An adversary ($\mathcal{A}$) is assumed to have intercepted messages Msg_1, Msg_2, Msg_3, and Msg_4 exchanged between the *6LN* and *6LS*. In the subsequent step, these messages take some time to reach their recipients after they are archived. Nevertheless, Msg_1 incorporates timestamp *T1*, Msg_2 incorporates timestamp *T2*, and Msg_3 incorporates timestamp *T3*. As soon as the *6LS* receives Msg_3, it confirms whether $|Tr-T2|$ and $|Tr-T1|$ are less than or equal to $T\Delta$. Similarly, when Msg_4 is received, the *6LN* checks whether $|Tr-T3|$ is equal to $T\Delta$. Moreover, the random numbers *RN1*, *RN2*, *R1*, and *R2* provide freshness to the messages. When the freshness tests fail for these messages, both sessions are terminated. Therefore, SLAE6 is protected against replay attacks.

Theorem 5: *SLAE6 is protected against DoS attacks.*
Proof: By spoofing IP addresses, IP spoofing attacks send large data packets over networks to launch a DoS attack. In SLAE6, a DoS attack requires an adversary ($\mathcal{A}$) to calculate the following: compute (*AD*), $M_{6LN}=(R1||TID_{6LN})$, $Si=(load\_AE(Ns||Km)||M_{6LN})$, and $(CT,Tag_{6LN})=E_{Si}\{AD,(A||B)\}$ to check the condition $Tag_d=Tag_{6LN}$. The condition $Tag_d=Tag_{6LN}$ will not hold because it requires parameters that are secret to the *6LN* and *6LS*, namely, ID_{6LN}, O_{SP}, and K_{6LN}. Hence, SLAE6 is protected against DoS attacks.

Theorem 6: *SLAE6 is protected against impersonation threats.*
***Proof*:** In this attack, an adversary ($\mathcal{A}$) attempts to mask as a legitimate *6LN* or *6LS*. It requires the construction of a valid acknowledgment message, Msg_1 and Msg_3. An impersonation attack can be prevented by considering the following cases.

6LN impersonation attack: $\mathcal{A}$ must produce a valid message Msg_1:$(T1||D||(CT||Tag_{6LN})||R1)$ on behalf of the *6LN* to execute an impersonation attack. $\mathcal{A}$ can easily generate a timestamp. Therefore, to generate the other parameters of a valid Msg_1, $\mathcal{A}$ must know the secret credentials, which include *D* and *CT*. The *6LN* knows only a few parameters. Therefore, SLAE6 is resistant to *6LN* impersonation.

6LS impersonation attack: The primary objective of this attack is to fool the *6LS* into believing that Msg_1 and Msg_2 are from the *6LN* by concocting Msg_3:$(ID_{6LBR}||T2\ ||Msg_2||S_{6LBR})$ on behalf of the *6LS*. The timestamps can be generated easily by $\mathcal{A}$. Only the *6LS* and *6LN* know the secret parameters, including ID_{6LN}, *CT*, and TID_{6LN}, which enable $\mathcal{A}$ to generate the remaining Msg_1. Creating a valid Msg_1 without these secret credentials is impossible, and thus, SLAE6 is immune to *6LS* impersonation attacks.

Theorem 7: *SLAE6 is robust against MITM attacks.*
***Proof*:** This attack is designed to intercept and modify messages Msg_1, Msg_2, Msg_3, and Msg_4. Then,

unsuspecting recipients receive these messages. Assume that an adversary ($\mathcal{A}$) captures all messages transmitted by Msg_1, Msg_2, Msg_3, and Msg_4 as the *6LN* communicates with the *6LS*. For example, suppose $\mathcal{A}$ forges Msg_1 for the *6LS* to believe that it is authentic. $\mathcal{A}$ must guess the real identity of the *6LN*, which is impossible. Therefore, $\mathcal{A}$ cannot generate a false message Msg_1. All other transmitted messages are subject to the same condition. Clearly, SLAE6 is protected against MITM attacks.

Theorem 8: *SLAE6 is resistant to ESL attacks.*

Proof: ESL attacks under the CK adversary model assume that leaks session-dependent ephemeral values, such that adversary ($\mathcal{A}$) cannot decode session keys and long-term secrets. SLAE6 establishes a secure communication key between the *6LN* and *6LS* during the AKE process. Several ephemeral terms, such as *RN1 and RN2*, and long terms, such as ID_{6LN}, are incorporated into the established *SK*. Even if A compromises RN1 and RN2, A still needs long-term TID_{6LN} to break SK's security SK=H (ID_{6LN}|| $B1$|| O_{SP1} || $RN1$|| $RN2$). *SK's* security cannot be compromised unless $\mathcal{A}$ knows what valid long and ephemeral terms are required. As a result, the proposed SLAE6 can withstand an ESL attack.

5.1.2 Ban Logic Analysis

An AKE process of SLAE6 is formally tested using BAN logic (Burrows et al., 1990) and determining whether participant agreements are trustworthy. To assess SLAE6's MA properties, BAN logic is used. BAN logic is described using the notations in Table 3, which describe how different inference rules are drawn. In Table 4, several logical rules for determining the goal of a proposed scheme are presented. SLAE6 makes the following assumptions as a starting point for investigating our scheme's AKE properties.

Table 3: BAN logic notations.

Feature	Description
$A \mid\equiv B$	*A* believes *B*.
$A \mid\sim B$	*A* once said *B*.
$A \lhd B$	*A* controls *B*.
$A \overset{k}{\leftrightarrow} R$	*E* and *R* share the key with *k*.
$A \overset{k}{\Leftrightarrow} R$	*k* is a secret parameter known only by *E* and *R*.
$\#(B)$	*B* is fresh.
$\{B\}k$	*B* is encrypted by *k*.
$(B)\,S$	*B* is combined with secret *S*.
$A \Rightarrow B$	*A* receives *B*.
$\frac{A}{R}$	If *A* is true, then *R* is also true.

Goals: BAN logic in SLAE6 is based largely on establishing an *SK* with each principal. As defined in Table 5, SLAE6 seeks to achieve the following goals for MA.

Table 4: BAN logic inference rules.

Notation	Description
Message Meaning Rule (MMR)	$\frac{A\mid\equiv A \overset{K}{\leftrightarrow} R,\, A \lhd \{B\}_K}{A\mid\equiv R \sim B}$
Jurisdiction Rule (JR)	$\frac{A\mid\equiv R \rightarrow B,\, A\mid\equiv R\mid\equiv B}{A\mid\equiv B}$
Belief Rule (BR)	$\frac{A\mid\equiv (B,R)}{A\mid\equiv B}$
Nonce Verification Rule (NVR)	$\frac{A\mid\equiv \#(R),\, A\mid\equiv R \sim B}{A\mid\equiv B}$
Freshness Rule (FR)	$\frac{A\mid\equiv \#(R)}{A\mid\equiv \#(R,B)}$

Table 5: Security goals.

No.	Goals
Goal 1	$6LS\mid\equiv 6LN\mid\equiv(6LN \overset{O_{SP}}{\longleftrightarrow} 6LS)$
Goal 2	$6LN\mid\equiv 6LN \overset{O_{SP}}{\longleftrightarrow} 6LS$
Goal 3	$6LS\mid\equiv 6LN\mid\equiv (6LS \overset{SK}{\leftrightarrow} 6LN)$
Goal 4	$6LS \overset{SK}{\leftrightarrow} 6LN$

Idealized Forms: Messages Msg_1, Msg_2, Msg_3, and Msg_4 sent by SLAE6 are transmitted on a public channel. Given their idealized form, these messages allow us to omit messages that do not provide the properties of BAN logic. Table 6 presents an idealized exchange of messages provided by SLAE6.

Table 6: Idealized message exchanges.

No.	*Msgs*
F1	$6LN \rightarrow 6LS$: $(T1, \{O_{SP}, RN1\})\, ID_{6LN})$
F2	$6LS \rightarrow 6LN$: $(T1, B1, \{O_{SP1}, RN2, (6LS \overset{SK}{\leftrightarrow} 6LN)\}\, ID_{6LN})$

Assumptions: At the end of registration, each principal is supposed to have an *SK*. After completing the registration process, the pseudo identities appear to be authentic and are random numbers. The entitlement components are also believed to be controlled by a legal principle, and SLAE6's BAN logic considers these assumptions in Table 7.

Table 7: Preliminary state assumptions.

No.	Goals
A1	$6LN \mid\equiv \#(T1), \#(T3)$
A2	$6LS \mid\equiv \#(T1), \#(T3)$
A3	$6LS \mid\equiv ID_{6LN}$
A4	$6LS \mid\equiv O_{SP}$
A5	$6LN \mid\equiv ID_{6LN}$
A6	$6LN \mid\equiv O_{SP}$
A7	$6LS \mid\equiv 6LS \overset{ID_{6LN}}{\longleftrightarrow} 6LN$
A8	$6LN \mid\equiv 6LS \overset{ID_{6LN}}{\longleftrightarrow} 6LN$
A9	$6LN \mid\equiv \#(RN_{S1})$
A10	$6LS \mid\equiv \#(RN_{S2})$
A11	$6LN \mid\equiv 6LS \Rightarrow (6LN \overset{ID_{6LN}}{\longleftrightarrow} 6LS)$
A12	$6LN \mid\equiv 6LS \Rightarrow (6LN \overset{SK}{\leftrightarrow} 6LS)$
A13	$6LN \mid\equiv 6LS \Rightarrow (6LN \overset{O_{SP}}{\longleftrightarrow} 6LS)$
A14	$6LS \mid\equiv 6LS \overset{O_{SP}}{\longleftrightarrow} 6LN$
A15	$6LN \mid\equiv 6LS \overset{O_{SP}}{\longleftrightarrow} 6LN$

BAN Logic Proof: To analyze SLAE6's BAN logic, the following steps are taken.

Step 1: From A7, A8, and F1 and by applying MMR, the following can be obtained:

$$S1 = \frac{6LS \mid\equiv \left(6LS \overset{ID_{6LN}}{\longleftrightarrow} 6LN\right), 6LS \triangleleft (T_{6LN}, \{O_{SP}, RN1\}) ID_{6LN}}{6LS \mid\equiv 6LN \sim (T_{6LN}, \{O_{SP}, RN1\}) ID_{6LN}).}$$

Step 2: S2 can be elicited by applying FR while using A2 and F1.

$$S2 = \frac{6LS \mid\equiv \#(T_{6LN})}{6LS \mid\equiv \#(T_{6LN}, \{O_{SP}, RN1\})}$$

Step 3: S3 can be elicited by applying NVR by using S2 and S1.

$$S3 = \frac{6LS \mid\equiv \#(T_{6LN}, \{O_{SP}, RN1\}), 6LS \mid\equiv 6LN \mid, A*}{6LS \mid\equiv 6LN \mid\equiv (T_{6LN}, \{O_{SP}, RN1\})}$$
$$A* = \sim (T_{6LN}, \{O_{SP}, RN1\})$$

Step 4: S4 can elicit Goal 1 by applying BR.

$$S4 = \frac{6LS \mid\equiv 6LN \mid\equiv (T_{6LN}, \{O_{SP}, RN1\})}{6LS \mid\equiv 6LN \mid\equiv (6LN \overset{O_{SP}}{\longleftrightarrow} 6LS))}$$

Step 5: Goal 2 can be achieved using A13, S4, and JR.

$$S5 = \frac{6LS \mid\equiv 6LN \mid \rightarrow \left(6LN \overset{O_{SP}}{\longleftrightarrow} 6LS\right) 6LS \mid * A}{6LS \mid\equiv (6LN \overset{O_{SP}}{\longleftrightarrow} 6LS)}$$
$$*A = \equiv 6LN \mid\equiv (6LN \overset{O_{SP}}{\longleftrightarrow} 6LS)$$

Step 6: S6 can be elicited by applying MMR by using F2 and A11.

1

$$S6 = \frac{6LS \mid\equiv (6LN \overset{ID_{6LN}}{\longleftrightarrow} 6LS), A*}{6LS \mid\equiv 6LN \mid \sim (T3, B1, \{O_{SP}, RN1, (6LN \overset{ID_{6LN}}{\longleftrightarrow} 6LS\}) ID_{6LN})}$$
$$A* = 6LS \triangleleft ((T3, B1, \{O_{SP}, RN1, (6LN \overset{ID_{6LN}}{\longleftrightarrow} 6LS\}) ID_{6LN})$$

Step 7: S7 can be elicited by applying FR by using A1 and F2.

$$S7 = \frac{6LS \mid\equiv \#(T3)}{6LS \mid\equiv \#(B1, \{O_{SP1}, RN1(6LN \overset{SK}{\leftrightarrow} 6LS))}$$

Step 8: S8 can be elicited by applying NVR by using S6 and S7.

$$S8 = \frac{6LN \mid\equiv \#(T3, B1, \{O_{SP}, RN1, (6LN \overset{SK}{\leftrightarrow} 6LS), A*}{6LN \mid\equiv 6LN \mid \sim (T3, B1, \{O_{SP}, RN1, (6LN \overset{ID_{6LN}}{\longleftrightarrow} 6LS\})}$$
$$A* = (6LN \overset{ID_{6LN}}{\longleftrightarrow} 6LS)\}), 6LN \mid\equiv 6LS \mid$$
$$\sim (T3, B1, \{O_{SP}, RN1, (6L \overset{ID_{6LN}}{\longleftrightarrow} 6LS)\})$$

Step 9: S9 obtains Goal 3 by employing BR by using S8.

$$S9 = \frac{6LN \mid\equiv 6LS(T3, B1, \{O_{SP}, RN1, (6LN \overset{SK}{\leftrightarrow} 6LS)\}}{6LN \mid\equiv 6LS \mid\equiv (6LN \overset{SK}{\leftrightarrow} 6LS)}$$

Step 10: S10 can elicit Goal 4 by applying JR by using A12.

$$S10 = \frac{6LN \mid\equiv \mid 6LN \mid \rightarrow \left(6LN \overset{SK}{\leftrightarrow} 6LS\right), A*}{6LN \mid\equiv (6LN \overset{SK}{\leftrightarrow} 6LS))}$$
$$A* = 6LN \mid\equiv 6LS \mid\equiv (T3, B1, \{O_{SP}, RN1, (6LN \overset{SK}{\leftrightarrow} 6LS)\}$$

As a result of Goals 1–4, we demonstrate that SLAE6 provides secure MA for the *6LN* and *6LS*. The MA properties of SLAE6 are assessed using BAN logic.

5.2 Performance Evaluation

The performance of an authentication protocol is evaluated by measuring its security features, computation cost, execution time and bandwidth requirements. SLAE6 is evaluated using the two metrics. We compare SLAE6 with the schemes in (Hussen et al., 2013), (Qiu & Ma, 2016), and (Tanveer et al., 2020).

5.2.1 Security Feature

The results in Table 8 indicate that SLAE6 is the only system that is capable of providing all security features. With the SLAE6 scheme, for example, sensor nodes can be protected from traceability attacks, which other authentication schemes do not

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

support. In addition, SLAE6 provides security under the CK model.

Table 8: Comparison of security features.

Features	(Hussen et al., 2013)	(Qiu & Ma, 2016)	(Tanveer et al., 2020)	SLAE6
Replay attack	YES	YES	YES	YES
ESL	NO	YES	YES	YES
untraceability	NO	NO	NO	YES
Anonymity	NO	YES	YES	YES
F/B S	NO	YES	YES	YES
Impersonation attack	YES	YES	YES	YES
DoS attack	YES	YES	YES	YES
MITM attack	YES	YES	YES	YES
MA	YES	YES	YES	YES
CK model	NO	NO	NO	YES

YES: security feature is supported/No: security feature is not supported.

5.2.2 Computational Complexity

This section compares the computation costs of SLAE6 with those of related authentication schemes, namely, (Hussen et al., 2013), (Qiu & Ma, 2016), and (Tanveer et al., 2020). This feature is calculated for all authentication schemes on the basis of the operations performed within each authentication entity. Cryptographic functions are executed by authentication nodes on the basis of the number of cryptographic functions that they have executed. This feature is used in all authentication schemes to facilitate computations. Denote the time cost T_{ACE}, T_H, T_{AES}, T_{ASCON}, T_{EXP}, T_{ECC}, and $T_{ECC/SV}$ to represent execution time for ACE, SHA-256, AES, ASCON, modular exponentiation, ECC key generation, and ECC signature generation/verification, respectively. Table 7 provides the computation cost in each authentication entity and the total computation cost of each authentication scheme. In accordance with Table 9, SLAE6 consumes less computational overhead than other existing schemes (Hussen et al., 2013), (Qiu & Ma, 2016), (Tanveer et al., 2020). The SLAE6 scheme argues that it is lightweight because it uses only symmetric encryption/decryption, XOR, and one-way HFs, which exhibit less computational complexity. Compared with existing authentication methods (Hussen et al., 2013), (Qiu & Ma, 2016), which use asymmetric encryption/decryption functions that are highly expensive, the proposed SLAE6 scheme takes nine HFs overhead during AKA phases, while the scheme of (Tanveer et al., 2020) takes 13 HFs.

5.2.3 Execution Time

Table 9 compares the execution times of the proposed protocols. We calculate execution time on the basis of the assumptions presented by Tanveer et al. (Tanveer et al., 2022) (Tanveer et al., 2020). $T_{ACE} \approx 0.0411$ ms, $T_H \approx 0.0311$ ms, $T_{AES} \approx 0.125$ ms, $T_{ASCON} \approx 0.065$ ms, $T_{EXP} \approx 19.16$ ms, $T_{ECC} \approx 5.50$, and $T_{SG} \approx 5.20$ ms. Consequently, during the AKE phase, $4T_{ACE} + 9T_H \approx 0.5159$ ms, the total cryptographic complexity is executed in SLAE6. Accordingly, for the protocols in (Hussen et al., 2013), (Qiu & Ma, 2016), and (Tanveer et al., 2020), the computation overheads are 58.6044, 17.2494, and 0.6643 ms, respectively, as illustrated in Table 9 and Figure 3. Therefore, the protocols in (Hussen et al., 2013) have a longer execution time of 58.6044 ms, followed by the protocols in (Qiu & Ma, 2016) and (Tanveer et al., 2020). Consequently, SLAE6 requires less execution time with 0.5159 ms.

Table 9: Comparison of the computational complexity and execution time.

Schemes	Computational Cost	Total Time
(Hussen et al., 2013)	$3T_{mx} + 8T_{AES} + 4T_{SHA\_256}$	58.6044 ms
(Qiu & Ma, 2016)	$5T_{AES} + 4T_{SHA\_256} + 2T_{ppk} + T_{sg}$	17.2494 ms
(Tanveer et al., 2020)	$4T_{ASCON} + 13T_{SHA\_256}$	0.6643 ms
SLAE6	$4T_{ACE} + 9T_H$	0.5159 ms

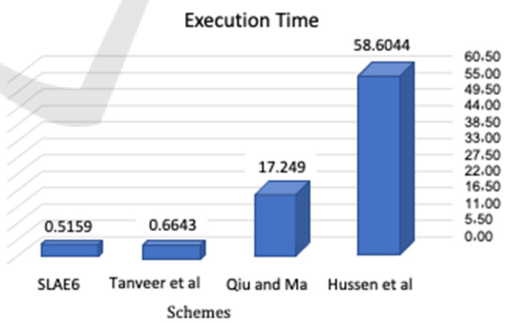

Figure 3: Comparison of execution time.

5.2.4 Bandwidth Requirements

This section estimates the bandwidth requirement for the AKE phase on the basis of the sizes of Msg_1 and Msg_4. It takes 256 bits for HF (SHA-256); 64 bits for a random number, identity, temerity identity, and secret parameters; 32 bits for timestamp; 160 bits for ECC; and 128 bits for ACE encryption and decryption. These messages are sized on the basis of

Table 8, which shows the sizes of the output from various cryptographic operations. On the basis of these messages, we can derive their sizes. Sensor nodes must minimize their transmitted message size to reduce their energy consumption. Notably, the *6LBR*, *6LS*, and *6LR* are energy-efficient devices. Therefore, power consumption outside 6LoWPAN is not considered in the SLAE6 model, because it focuses only on power consumption of wirelessly connected constrained devices. Various bandwidth requirements between the *6LN* and *6LR* are presented in Table 10 and shown in Figure 4. This table shows that the scheme in (Hussen et al., 2013) requires a maximum bandwidth of 2864 bits. This scheme is followed by the schemes in (Qiu & Ma, 2016) and (Tanveer et al., 2020). By contrast, SLAE6 requires only 850 bits of bandwidth.

Table 10: Bandwidth requirements.

Exchanged messages	*6LN*→*6LR*	*6LR*→*6LN*	Total messages
(Hussen et al., 2013)	688 bits	2176 bits	2864 bits
(Qiu & Ma, 2016)	672 bits	784 bits	1456 bits
(Tanveer et al., 2020)	496 bits	528 bits	1024 bits
SLAE6	425 bits	425 bits	850 bits

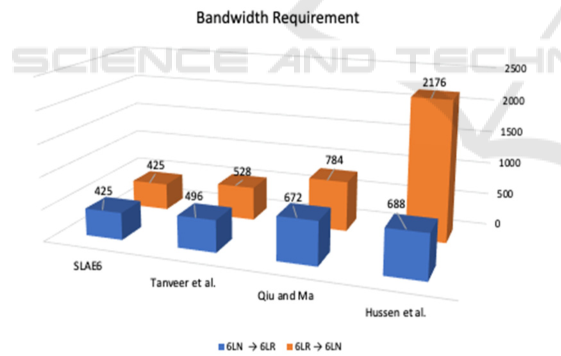

Figure 4: Comparison of bandwidth requirement.

6 CONCLUSIONS AND FUTURE WORK

Conventional ECC, PKI, signature, and identity-based AKE schemes generate high communication and computation overheads that are inappropriate for limited-resource 6LoWPAN devices. Therefore, we proposed the SLAE6 device AKE scheme in the current study. This scheme, which is based on the ACE cryptographic mechanism via LWC, is efficient and secure. MA is performed in SLAE6, and a secure connection is established between SNs and the server for encrypted communication. This process ensured secure communication and prevented an attacker from obtaining transmitted information. Hence, energy consumption and cost minimization are achieved while ensuring information security. BAN logic analysis proves that SLAE6 is logically complete. In addition, SLAE6 is proven robust against known attacks by using informal security analysis based on the DY and CK models. We present a verifiable security and privacy provisioning protocol designed to address some of these issues in the current study. SLAE6 exhibits less computation complexity and effectively reduces execution time by 22% and requirement bandwidths by 16% compared with (Tanveer et al., 2020). Consequently, this protocol has been demonstrated to be efficient in terms of bandwidth usage and execution time. It is computationally inexpensive and suited for SNs with limited resources in IoT or WSN. In the future, this protocol needs to be formally verified. Then, it will be applied to a test bed experiment.

ACKNOWLEDGMENT

This research was supported by the research grant of the Universiti Kebangsaan Malaysia under FRGS grant: FRGS/1/2022/ICT11/UKM/02/2.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

REFERENCES

Aagaard, M., Altawy, R., Gong, G., Mandal, K., Rohit, R., & Lab, S. (2019). ACE: An Authenticated Encryption and Hash Algorithm Submission to the NIST LWC Competition. https://uwaterloo.ca/communications-security-lab/lwc/ace

Ahmad, R., Sundarajan, E. A., & Abu-Ain, T. (2021). Analysis the Effect of Dynamic Clustering and Lightweight Symmetric Encryption Approaches on Network Lifetime in WSNs. Iceei2021, 1–28.

Ahmad, R., Wazirali, R., Abu-Ain, T., & Almohamad, T. A. (2022). Adaptive Trust-Based Framework for Securing and Reducing Cost in Low-Cost 6LoWPAN Wireless Sensor Networks. Applied Sciences, 12(17), 8605. https://doi.org/10.3390/app12178605

Ahmad, R., Wazirali, R., Bsoul, Q., Abu-Ain, T., & Abu-Ain, W. (2021). Feature-selection and mutual-clustering approaches to improve dos detection and maintain wsns' lifetime. Sensors, 21(14). https://doi.org/10.3390/s21144821

Alshrif, F. F., Sundararajan, E. A., & Ahmad, R. (2021). New Framework for Authentication and key

Establishment to Secure 6LoWPAN Networks. Iceei2021, 1–6.
Amanlou, S., Hasan, M. K., & Bakar, K. A. A. (2021). Lightweight and secure authentication scheme for IoT network based on publish–subscribe fog computing model. Computer Networks, 199. https://doi.org/10.1016/j.comnet.2021.108465
Bagwari, A., Bagwari, J., Anand, T., Chaurasia, B. K., Gangwar, R. P. S., & Hasan, M. K. (2022). The Role of IoT in Smart Technologies. In 5G and Beyond (pp. 57–66). Chapman and Hall/CRC. https://doi.org/10.1201/9781003045809-6
Burrows, M., Abadi, M., & Needham, R. (1990). Logic of authentication. ACM Transactions on Computer Systems, 8(1), 18–36. https://doi.org/10.1145/77648.77649
Chom Thungon, L., Ahmed, N., Chandra Sahana, S., & Hussain, M. I. (2020). A lightweight authentication and key exchange mechanism for IPv6 over low-power wireless personal area networks-based Internet of things. Transactions on Emerging Telecommunications Technologies, 1–18. https://doi.org/10.1002/ett.4033
Dolev, D., & Yao, A. C. (1983). On the Security of Public Key Protocols. IEEE Transactions on Information Theory, 29(2), 198–208. https://doi.org/10.1109/TIT.1983.1056650
Fragkos, G., Minwalla, C., Tsiropoulou, E. E., & Plusquellic, J. (2022). Enhancing Privacy in PUF-Cash through Multiple Trusted Third Parties and Reinforcement Learning. ACM Journal on Emerging Technologies in Computing Systems, 18(1). https://doi.org/10.1145/3441139
Gao, L., Zhang, L., Feng, L., & Ma, M. (2020). An Efficient Secure Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN in Unattended Scenarios. Wireless Personal Communications, 115(2), 1603–1621.https://doi.org/10.1007/s11277-020-07645-z
Hasan, M. K., Shafiq, M., Islam, S., Pandey, B., Baker El-Ebiary, Y. A., Nafi, N. S., Ciro Rodriguez, R., & Vargas, D. E. (2021). Lightweight Cryptographic Algorithms for Guessing Attack Protection in Complex Internet of Things Applications. Complexity, 2021. https://doi.org/10.1155/2021/5540296
Hennebert, C., & Santos, J. Dos. (2014). Security protocols and privacy issues into 6LoWPAN stack: A synthesis. IEEE Internet of Things Journal, 1(5), 384–398. https://doi.org/10.1109/JIOT.2014.2359538
Hui, J., & Pascal, T. (2011, August 20). Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. 2011. https://www.hjp.at/doc/rfc/rfc6282.html
Hussen, H. R., Tizazu, G. A., Ting, M., Lee, T., Choi, Y., & Kim, K.-H. (2013). SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6L0WPAN). In 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE., 246–251. https://doi.org/https://doi.org/10.1109/ICUFN.2013.6614820
Monika, M., Smita, A., Rakhi, A., Sanjivani, D., Sachi, N. M., & Suneeta, S. (2022). A Proposed Framework to Achieve CIA in IoT Networks. International Conference on Artificial Intelligence and Sustainable Engineering, 19–30.
Qiu, Y., & Ma, M. (2016). A Mutual Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN Networks. IEEE Transactions on Industrial Informatics, 12(6), 2074–2085. https://doi.org/10.1109/TII.2016.2604681
Sarr, A. P., Elbaz-Vincent, P., & Bajard, J.-C. (2010). A New Security Model for Authenticated Key Agreement. International Conference on Security and Cryptography for Networks, 219–234. https://doi.org/10.1007/978-3-642-15317-4_15
Tanaka, Y., Minet, P., & Watteyne, T. (2019). 6LoWPAN Fragment Forwarding. IEEE Communications Standards Magazine, 3(1), 35–39. https://doi.org/10.1109/MCOMSTD.2019.1800029
Tanveer, M., Abbas, G., Abbas, Z. H., Bilal, M., Mukherjee, A., & Kwak, K. S. (2021). LAKE-6SH: Lightweight User Authenticated Key Exchange for 6LoWPAN-based Smart Homes. IEEE Internet of Things Journal, 14(8), 1–14. https://doi.org/10.1109/jiot.2021.3085595
Tanveer, M., Abbas, G., Abbas, Z. H., Waqas, M., Muhammad, F., & Kim, S. (2020). S6AE: Securing 6lowpan using authenticated encryption scheme. Sensors (Switzerland), 20(9), 1–23. https://doi.org/10.3390/s20092707
Tanveer, M., Khan, A. U., Nguyen, T., Ahmad, M., & Abdei-Latif, A. (2022). Towards A Secure and Computational Framework for Internet of Drones Enabled Aerial Computing. IEEE Transactions on Network Science and Engineering. https://doi.org/10.1109/TNSE.2022.3151843
Turan, M., McKay, K., Chang, D., Calik, C., Bassham, L., Kang, J., & Kelsey, J. (2021). Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process. https://doi.org/10.6028/NIST.IR.8369
Wazirali, R., & Ahmad, R. (2022). Machine learning approaches to detect DoS and their effect on WSNs lifetime. Computers, Materials and Continua, 70(3), 4921–4946. https://doi.org/10.32604/cmc.2022.020044
Wazirali, R., Ahmad, R., Al-Amayreh, A., Al-Madi, M., & Khalifeh, A. (2021). Secure watermarking schemes and their approaches in the iot technology: An overview. Electronics (Switzerland), 10(14). https://doi.org/10.3390/electronics10141744