Received: 00 Month 0000 | Revised: 00 Month 0000 | Accepted: 00 Month 0000

DOI: xxx/xxxx

ARTICLE TYPE

Testbed Evaluation of Light weight Authentication Protocol(LAUP) for 6LoWPAN wireless sensor networks

Annie Gilda Roselin[1,2] | Priyadarsi Nanda[1] | Surya Nepal[2] | Sean He[1]

[1]School of Electrical and Data Engineering, University of Technology Sydney(UTS), Sydney, Australia
[2]CSIRO, Data61,Marsfield, NSW, Australia

Correspondence
Annie Gilda Roselin, Email:
Annie.G.ArockiaBaskaran@student.uts.edu.au

Summary

6LoWPAN networks involving wireless sensors consist of resource starving miniature sensor nodes. Since secured authentication is one of the important considerations, use of asymmetric key distribution scheme may not be a perfect choice. Recent research shows that Lucky Thirteen attack has compromised Datagram Transport Layer Security (DTLS) with Cipher Block Chaining (CBC) mode for key establishment. Even though EAKES6Lo and S3K techniques for key establishment follow the symmetric key establishment method, they strongly rely on a remote server and trust anchor. Our proposed Lightweight Authentication Protocol (LAUP) used a symmetric key method with no preshared keys and comprised of four flights to establish authentication and session key distribution between sensors and Edge Router in a 6LoWPAN environment. Each flight uses freshly derived keys from existing information such as PAN ID (Personal Area Network IDentification) and device identities. We formally verified our scheme using the Scyther security protocol verification tool. We simulated and evaluated the proposed LAUP protocol using COOJA simulator and achieved less computational time and low power consumption compared to existing authentication protocols such as the EAKES6Lo and SAKES. And LAUP is evaluated using real-time test bed and achieved less computational time which is supportive of our simulated results.

KEYWORDS:
6LoWPAN, Authentication, Session key

1 | INTRODUCTION

The network of low power sensors called LoWPAN (Low Power Wireless Personal Area Network) consists of small sensors with limited memory, less computational capability and low in resources. 6LoWPAN is one of the most important communication protocols used in LoWPAN network. 6LoWPAN is designed in such a way that it can transmit the IPv6 packets over LoWPAN network (1)(2)(3) (4). Security in LoWPAN is provided through authentication of sensor device before any communication happens.

Last MAC and Handshake Authentication methods are mainly used to provide anti-replay protection and authentication (5) for low power devices. The MAC value (hashed value of the message and key) of the previous message is appended to the current message, and the receiver validates the received MAC with the already stored MAC value. However, this technique assumes that the receiver will always accept the first message packet and globally shared keys for authentication. Even though ECC based Diffie-Hellman key exchange with Kerberos authentication (6) for wireless sensor networks provide high security, symmetric encryption and handshake authentication are the highly desirable mechanisms to build the authentication for the successive message transmissions of LoWPAN devices. The public key method of key distribution requires more computational time and energy which leads to overhead for lightweight sensors.

TABLE 1 Theoretical comparison and Features of LAUP

No	Parameters	Compressed DTLS	EAKES6Lo	S3K	LAUP
1	Asymmetric method	yes	No	No	No
2	Symmetric + preshared keys	No	Yes	Yes	No
3	Use of Key Distribution centre	No	Yes	Yes	No
4	Symmetric + No preshared keys	No	No	No	Yes

The plaintext of LoWPAN communication can be recovered from a DTLS connection using an OpenSSL implementation of DTLS when using CBC (Cipher Block Chaining) mode encryption (7). So it is not a good idea to have DTLS for Key management otherwise DTLS-CBC mode has to be enhanced to manage Lucky thirteen type attacks. To overcome these attacks, LAUP uses ECB (Electronic Code Book) mode of AES-128 encryption. Moreover, compressed DTLS has six flights for authentication, whereas our proposed LAUP has only four flights for authentication and session key distribution. Existing mechanisms such as lightweight IKEv2, EAKES6Lo, S3K and compressed DTLS are providing lightweight authentication for wireless sensor communication. Lightweight IKEv2 (8) is secure but requires more memory for calculation. EAKES6Lo (9) uses hash functions to ensure the integrity of messages. Though this technique uses symmetric key cryptography, it assumes that the secret key distributed to every node by the remote server. The distribution of secret keys remains a challenge.

S3K symmetric key establishment for IoT (8) uses trust anchor and a resource server for key distribution among clients. However, the secret key is shared between trust anchor and resource server before deployment. The limitation of this idea is that it is not scalable for a vast network and mobility nodes. Maintaining the uniqueness of shared key between trust anchor and resource server is complicated. S3K runs in addition to DTLS and CoAP protocols which in turn, increases the computational overhead of LoWPAN devices. Moreover, the secure connection has to be established between trust anchor and remote server before the key generation process using TLS/DTLS.

After a thorough study on key distribution and authentication of 6LoWPAN networks, we come to the following conclusions. The existing algorithms, which follow pre-shared key methods are facing problems while updating the keys and having the assumptions of pre-shared keys. Moreover, they are relying on a key distribution center (10) to distribute the keys which in turn cost more resources and providing protection to key distribution center are another problem in real time. Well established existing algorithms which are using asymmetric method for authentication and key distribution are having the limitation of spending more computational time for authentication. To overcome these existing constraints, we proposed our LAUP algorithm for authentication and key establishment. LAUP leads to the highly secured and easily adaptable authentication method for 6LoWPAN networks. Table 1 shows the explanation of our approach to 6LoWPAN networking.

Our proposed LAUP authentication algorithm addresses the significant challenges such as a distribution of preshared keys and usage of resource centers for key distribution in the field of authentication among low power devices. We made observations on how the conventional network protocols compressed to be compatible with LoWPAN Wireless Sensor Networks (WSN). Our observation showed that Compressed DTLS Handshake (11) uses six flights for authentication and key exchange, whereas our protocol uses only four flights for the same. To our knowledge, LAUP authentication algorithm is the most suitable for 6LoWPAN network and secure authentication algorithm without using preshared keys for authentication and key distribution of LoWPAN devices.

The rest of the paper is organized as follows. Section 2 explains related works in the area of authentication and key distribution of LoWPAN networks. Section 3 describes our proposed work using session request phase, authentication phase, and key distribution phase. Formal verification using Scyther tool is presented in section 4. Section 5 analyses the efficiency of the LAUP protocol against various attacks of LoWPAN network. Performance evaluation of LAUP using Contiki OS COOJA simulator and a real-time test bed is demonstrated in section 6. Finally, we conclude the paper in section 7.

2 | RELATED WORK

APKES (Adaptive Pairwise Key Establishment Scheme) (12) uses pre-distributed pairwise keys to derive pairwise session keys for authentication. SPINS (Security Protocol for Sensor Networks) (13) scheme provides security for wireless sensor communication also obtains keys from the pre-distributed master keys. Although AKES (Adaptive Key Establishment Scheme) (14) system uses PAN ID and address of the sensor for authentication and key distribution, it follows pre-distribution of keys to derive pairwise session keys. Unlike our LAUP, AKES uses the address of sensors to get the shared secret keys, whereas LAUP uses MAC ID of sensors. Moreover, sensors (LoWPAN devices) which are using AKES scheme for authentication, to be preloaded with any of the relevant addressing information like 8-byte extended, 2-byte short or 1-byte simple address. Since

TABLE 2 Key Derivation Process

Flight No	Key	Process
1	SK1	PANID
2	SK2	ID_S XOR $Nonce_S$ XOR SK1
3	SK3	ID_{ER} XOR SK2
4	SK4	$Nonce_{S'}$ XOR $Nonce_S$ XOR SK3
SESSIONKEY	$K_{Session}$	SK1 XOR SK2 XOR SK3 XOR SK4 XOR $Nonce_{ER'}$

LAUP uses MAC ID of devices, it does not need any reloading of address. APKES, SPINS and AKES methods discussed previously, but in most cases, attention directed towards pre-distributed keys for key distribution and authentication.

SAKES (15) and EAKES6Lo authentication schemes deal with pre-shared keys among the 6LoWPAN host, 6LoWPAN router, and 6LoWPAN edge router. EAKES6Lo (9) has three phases such as pre-deployment phase, authentication and key establishment phase and handover phase. The remote server distributes private/public keys used by the sensors during authentication. A registration request by the sensor node is sent to the server with sensor ID and its public key. This public key is derived using ECDH (Elliptic Curve Diffie-Helman) mechanism which is more power consuming process for LoWPAN devices. But LAUP eliminates this additional power requirement by not using a remote server for preshared keys. LAUP focuses on preventing attacks on transportation layer such as replay attack, a man in the middle attack, and impersonation attack.

GDP (Group Device Pairing) does not need extra hardware devices for preshared keys. Based on symmetric key cryptographic techniques GDP provides secure communication between wireless body area networks. Even though GDP method (16) supports no redistribution of keys, GDP needs human user intervention for verification during authentication. Periodical updates of local keys (17) could prevent sensor compromisation on static nodes. Smaller cryptographic keys play a significant role in providing security for sensor communication (18). Unlike GDP, LAUP does not need human intervention during authentication and key establishment process. Moreover, LAUP session keys are small and have a periodical update for each session.

3 | PROPOSED WORK

Our proposed LAUP provides security to the 6LoWPAN device communication by authenticating the intended 6LoWPAN devices with Edge Router. 6LoWPAN is designed in such a way that it can transmit the IPv6 packets over Low Power Personal Area Network. 6LoWPAN protocol stack adopts bottom-most two layers from IEEE 802.15.4. 6LoWPAN acts as an adaptation layer between the link layer and the network layer. Figure 1(a) shows 6LoWPAN protocol stack and examples of protocols used in each layer. IEEE 802.15.4 supports only 127-byte packet length of messages. But the Maximum Transferrable Unit (MTU) of IPV6 is 1280 bytes. 6LoWPAN adaptation layer provides fragmentation and re-ordering, and compression of the protocol stack headers of IPv6 packets to maintain the communication compatibility between IEEE 802.15.4 frame and the legacy Internet message packet (1), (19), (20). LAUP works on transport layer of the 6LoWPAN protocol.

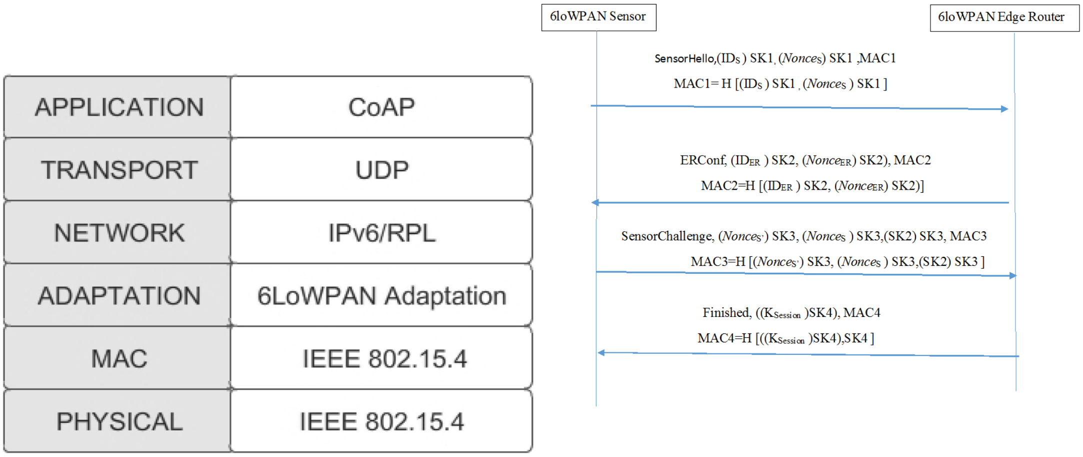
System architecture of LAUP is a hybrid of simple LoWPAN and an Ad-hoc LoWPAN architecture (1) shown in Figure 4(a). This architecture includes a 6LoWPAN device which is intended to communicate with the 6LoWPAN Edge Router and an Edge Router. LAUP works on the 6LoWPAN wireless sensor network communication between the 6LoWPAN device and the Edge Router. To maintain the secure communication 6LoWPAN device has to reside in the coverage range of Border Router.

3.1 | Basic assumptions of proposed work

Every sensor identity ID_S (MAC Address of sensor) is registered with the 6LoWPAN Edge Router ($6L_{ER}$) and they are physically secured. Our LAUP deals with 6LoWPAN devices which are deployed within the coverage range of 6LoWPAN Edge Router $6L_{ER}$. $6L_{ER}$ knows the PAN ID of LoWPAN network, and we assume that the sensors connected to the LoWPAN network are physically secured. We address the authentication and session key establishment of sensors when they are communicating to $6L_{ER}$. Each flight calculates its key for session key distribution by following common key derivation method. LAUP does not use any software or hardware based random number generation scheme to produce nonce values. Instead, the time of message generation on each flight has been taken as nonce respectively. By this way of receiving a nonce value, reduces the extra computational complexity and memory usage of low power devices. Table 2 explains the key derivation method to calculate unique flight keys. These methods use simple XOR functions with available values such as PAN ID, MAC ID and nonce values.

3.2 | Proposed LAUP Scheme

LAUP allows sensors of LoWPAN networks to communicate with a router to get cryptographically secure session key by two-level authentication using MAC ID of sensors and their nonce values. Figure 1(b) shows the process of LAUP communication between the 6LoWPAN sensor and an Edge Router.



APPLICATION	CoAP
TRANSPORT	UDP
NETWORK	IPv6/RPL
ADAPTATION	6LoWPAN Adaptation
MAC	IEEE 802.15.4
PHYSICAL	IEEE 802.15.4

6loWPAN Sensor
6loWPAN Edge Router
SensorHello,(IDS) SK1, (NonceS) SK1 ,MAC1
MAC1= H [(IDS) SK1 , (NonceS) SK1]
ERConf, (IDER) SK2, (NonceER) SK2), MAC2
MAC2=H [(IDER) SK2, (NonceER) SK2)]
SensorChallenge, (NonceS') SK3, (NonceS) SK3,(SK2) SK3, MAC3
MAC3=H [(NonceS') SK3, (NonceS) SK3,(SK2) SK3]
Finished, ((KSession)SK4), MAC4
MAC4=H [((KSession)SK4),SK4]

(a) 6LoWPAN protocol stack

(b) Proposed Authentication Algorithm

FIGURE 1 The 6LoWPAN protocol stack and the proposed LAUP algorithm.

LAUP algorithm gives protection against a Replay Attack, Man in the Middle attack, and impersonation attack by including the MAC values and nonce values. Even the attacker eavesdropped the message; he can not be able to reproduce the same message since the message is in the encrypted form and it needs the exact time of when the packet was generated. As a result of our LAUP algorithm, a unique session key will be produced by the Edge Router for a sensor claims **SensorHello** request. Figure 2 shows the flow of communication of LAUP on a 6LoWPAN device and the Edge Router. To encrypt the messages in each flight, we use the AES-128-ECB algorithm. A simple XOR function is used as a hash function to produce MAC values in all the four flights of communication. Our proposed LAUP algorithm has three phases (Session Request, Authentication and Key Distribution Phase) for authentication and session key establishment process.

3.2.1 | Session Request phase

Session request phase comprises of $flight_{one}$ communication message. In this phase, the 6LoWPAN sensor which is intended to communicate with the 6LoWPAN Edge Router $6L_{ER}$ sends the following content in its payload to the $6L_{ER}$. PAN ID of the network acts as a $flight_{one}$ key to encrypt the messages involved in first flight communication. The identity (MAC_ID) and the timer value (time generated by the sensor) of the sensor is encrypted by the $flight_{one}$ key called SK1. MAC_{one} value is calculated by applying the XOR function on the encrypted messages. Encrypted identity, encrypted nonce of the sensor, MAC_{one} value and "SensorHello" message are sent as a first flight information to $6L_{ER}$. Hence the identity and the nonce value of the sensor is retrieved by the $6L_{ER}$.

3.2.2 | Authentication phase

After receiving the $flight_{one}$ information from sensor, MAC_{one} value is calculated at Edge Router by hashing the received encrypted values. Before validating the authenticity of the sensor, the received MAC_{one} value is compared with the calculated MAC_{one} value. If both the values are same, protection against replay attack can be ensured, and the authentication process is going to be carried out by the Edge Router.

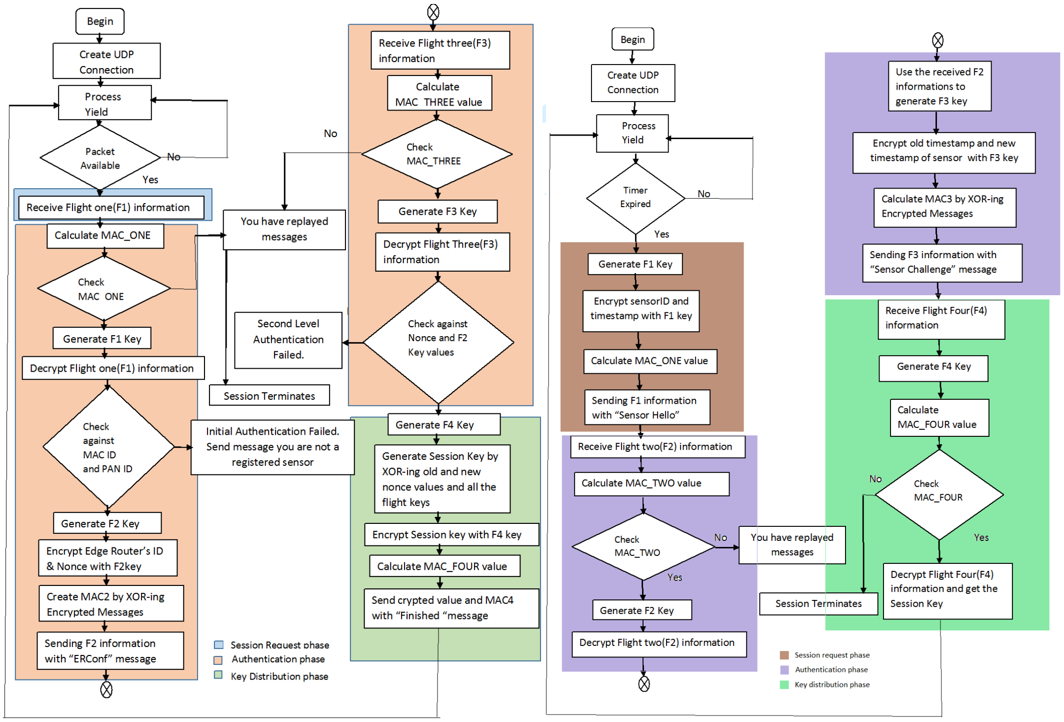
Two levels of authentication (Initial level and second level) will be performed by the $6L_{ER}$. Flow chart of Edge Router process in Figure 2 explains the two level authentication process in detail. In the Initial level of authentication, the received flight one information are decrypted using an AES-128 algorithm with ECB mode. Retrieved sensor MAC_ID from $flight_{one}$ information is checked against the already registered sensor MAC_ID. If a match is found, then $6L_{ER}$ generates $flight_{two}$ key SK2 by applying the XOR function on sensor `MAC_ID`, sensor nonce value and $flight_{one}$ key. Otherwise, 6LER send "You are not a registered sensor" message to the corresponding sensor and terminates the session.

$flight_{two}$ information is generated by the Edge Router and communicated with a sensor which claims authentication for session keys. Edge Router ID and nonce value is encrypted with $flight_{two}$ key SK2 and MAC_{two} is calculated by using a hash function on the resultant encrypted values. Along with the encrypted values and MAC_{two} values, "ERConf" message is sent to the sensor as a second flight information. After receiving $flight_{two}$ information, sensor checks the MAC_{two} value by calculating the same with the procedure followed by the Edge Router for MAC_{two} calculation. If the MAC_{two} is not replayed by any adversaries, then the sensor generates $flight_{two}$ key SK2 and decrypt the received $flight_{two}$ information from the Edge Router. After the above steps are performed, the sensor stores the ID of the Edge Router.

With the retrieved values from $flight_{two}$ packet, the sensor generates $flight_{three}$ key SK3 by applying XOR functions between Edge Router ID and $flight_{two}$ key SK2. $flight_{three}$ information are generated by encrypting the old and new nonce value of sensor and $flight_{two}$ key, with $flight_{three}$ key. Then MAC_{three} is calculated by applying a hash function on the encrypted values. Now $flight_{three}$ information packet is ready to send to Edge Router along with the "Sensor Challenge" message. When the Edge Router receives $flight_{three}$ information from sensor, Edge Router checks whether it has sent $flight_{two}$ information to the intended sensor.

Upon getting positive results of the checking operation, Edge Router starts to process the received $flight_{three}$ information from sensors. Initially Edge Router checks MAC_{three} value by calculating it and compare with the received MAC_{three} value. If the Edge Router found, the packet is not replayed, then proceed to calculate $flight_{three}$ key SK3 otherwise, terminates the session by sending "You have replayed the message".

$flight_{three}$ information are decrypted using $flight_{three}$ key SK3 and Edge Router gets the information like old, the new nonce value of sensor and $flight_{two}$ key SK2 calculated by the sensor. Edge Router does the second level authentication by comparing the nonce value of sensor what it has received from $flight_{two}$ and comparing $flight_{two}$ key SK2 value with the existing information. If the value matches, then it starts to process the further required session key generation. Thus the authentication phase of the sensor is completed by the Edge Router.



Begin
Create UDP Connection
Process Yield
Packet Available
No
Yes
Receive Flight one(F1) information
Calculate MAC_ONE
Check MAC ONE
Generate F1 Key
Decrypt Flight one(F1) information
Check against MAC ID and PAN ID
Generate F2 Key
Encrypt Edge Router's ID & Nonce with F2key
Create MAC2 by XOR-ing Encrypted Messages
Sending F2 information with "ERConf" message
You have replayed messages
Second Level Authentication Failed.
Session Terminates
Initial Authentication Failed. Send message you are not a registered sensor
No
Receive Flight three(F3) information
Calculate MAC THREE value
Check MAC_THREE
Generate F3 Key
Decrypt Flight Three(F3) information
Check against Nonce and F2 Key values
Generate F4 Key
Generate Session Key by XOR-ing old and new nonce values and all the flight keys
Encrypt Session key with F4 key
Calculate MAC_FOUR value
Send crypted value and MAC4 with "Finished "message
Session Request phase
Authentication phase
Key Distribution phase
Begin
Create UDP Connection
Process Yield
Timer Expired
No
Yes
Generate F1 Key
Encrypt sensorID and timestamp with F1 key
Calculate MAC_ONE value
Sending F1 information with "Sensor Hello"
Receive Flight two(F2) information
Calculate MAC_TWO value
Check MAC_TWO
No
Yes
You have replayed messages
Generate F2 Key
Decrypt Flight two(F2) information
Use the received F2 informations to generate F3 key
Encrypt old timestamp and new timestamp of sensor with F3 key
Calculate MAC3 by XOR-ing Encrypted Messages
Sending F3 information with "Sensor Challenge" message
Receive Flight Four(F4) information
Generate F4 Key
Calculate MAC_FOUR value
Check MAC_FOUR
No
Yes
Session Terminates
Decrypt Flight Four(F4) information and get the Session Key
Session request phase
Authentication phase
Key distribution phase

(a) Flow chart of Edge Router Process

(b) Flow chart of 6LoWPAN sensor Process

FIGURE 2 The process flow of LAUP on 6LoWPAN device and Edge Router.

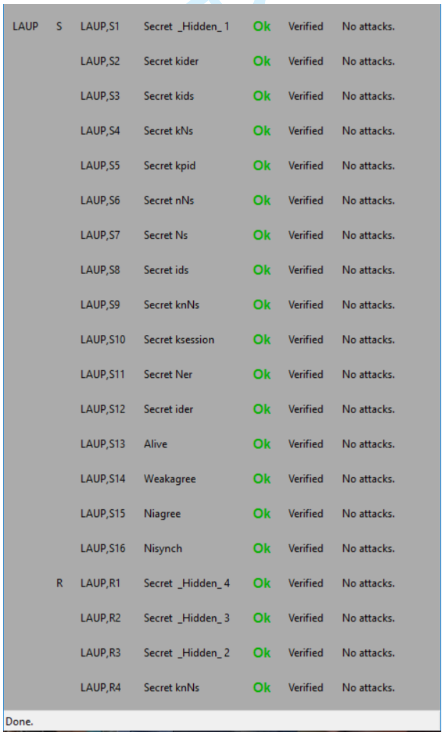
3.2.3 | Key Distribution phase

$flight_{four}$ key SK4 is the composition of old and new nonce values of the sensor and then the $flight_{three}$ key SK3. The session key ($K_{Session}$) is composed of $flight_{one}$ SK1, $flight_{two}$ SK2, $flight_{three}$ SK3, $flight_{four}$ SK4 and nonce of Edge Router at the time of session key generation. The session key is generated by the Edge Router by applying the XOR function on the above said values. The session key is encrypted with $flight_{four}$ key SK4 and is generated by the Edge Router. MAC_{four} value is calculated using XOR functions on encrypted values and $flight_{four}$ key SK4. After the cryptographic functions and MAC_{four} calculation, $flight_{four}$ information is sent to the intended sensor with the “Finished” message.

$flight_{four}$ information are received by the sensor and sensor generates $flight_{four}$ key SK4 using the same method followed by the Edge Router. A sensor checks whether the message is replayed or not by checking the MAC_{four} value with the calculated MAC_{four}. If the $flight_{four}$ message packet, through the checking operation of MAC values, then the sensor decrypts the $flight_{four}$ message and get the session key ($K_{Session}$). Thus the key distribution process is completely done by the Edge Router to the sensor by means of communicating four flight messages. This session key is used as a key to encrypt the further communication.

4 | FORMAL VERIFICATION OF LAUP ALGORITHM

The Scyther formal verification tool is used to verify the authentication properties of our proposed algorithm. Definitions of Aliveness, Secrecy, Non-Injective-Agreement, and Non-Injective-Synchronization are defined in (21, 22). Figure 3 shows that LAUP algorithm satisfies all the specified authentication properties such as aliveness, secrecy, non-injective-agreement, and non-injective-synchronization. We have proved the secrecy of Edge Router ID and Non-Injective Synchronization of LAUP based on (21, 23).



```
LAUP   S   LAUP,S1    Secret _Hidden_ 1   Ok   Verified   No attacks.
           LAUP,S2    Secret kider        Ok   Verified   No attacks.
           LAUP,S3    Secret kids         Ok   Verified   No attacks.
           LAUP,S4    Secret kNs          Ok   Verified   No attacks.
           LAUP,S5    Secret kpid         Ok   Verified   No attacks.
           LAUP,S6    Secret nNs          Ok   Verified   No attacks.
           LAUP,S7    Secret Ns           Ok   Verified   No attacks.
           LAUP,S8    Secret ids          Ok   Verified   No attacks.
           LAUP,S9    Secret knNs         Ok   Verified   No attacks.
           LAUP,S10   Secret ksession     Ok   Verified   No attacks.
           LAUP,S11   Secret Ner          Ok   Verified   No attacks.
           LAUP,S12   Secret ider         Ok   Verified   No attacks.
           LAUP,S13   Alive               Ok   Verified   No attacks.
           LAUP,S14   Weakagree           Ok   Verified   No attacks.
           LAUP,S15   Niagree             Ok   Verified   No attacks.
           LAUP,S16   Nisynch             Ok   Verified   No attacks.
       R   LAUP,R1    Secret _Hidden_ 4   Ok   Verified   No attacks.
           LAUP,R2    Secret _Hidden_ 3   Ok   Verified   No attacks.
           LAUP,R3    Secret _Hidden_ 2   Ok   Verified   No attacks.
           LAUP,R4    Secret knNs         Ok   Verified   No attacks.
Done.
```

FIGURE 3 Sycther tool results for LAUP Protocol verification

Secrecy: Secrecy expresses that certain information is not revealed to an intruder, even though we are communicating this data over an untrusted network. By maintaining the secrecy of edge router ID, we can perform second level authentication of 6LoWPAN devices with edge router also the intruder can not get any information of edge router ID.

Non-Injective Synchronization property of 6LoWPAN sensor, ensures that it communicates with the intended party 6LoWPAN Edge Router and the contents of receiving/sending messages are equal. Also, it guarantees the expected order of send and receives actions.

LAUP is specified as follows:

`LAUP(s)` $= \{s, r, (ID_{ER}, k1(s,r), k2(s,r), k3(s,r), k4(s,r)\},$
$send_1(s, r, \{|ID_S|\}k1(s,r), \{|TS_S|\}, m), read_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}, k2(s,r), m1),$
$send_3(s, r, \{|TS_{S'}|\}k3(s,r), \{|TS_S|\}, k3(s,r), \{|k2|\}k3(s,r)m2), read_4(r, s, Ksession, k4(s,r), m3),$
$claim_5(s, secret, ID_{ER}), claim_6(s, secret, TS_S), claim_7(s, nisynch))$
`LAUP(r)` $= (\{s, r, ID_S, TS_S, k1(s,r), k2(s,r), k3(s,r), k4(s,r)\},$
$read_1(s, r, \{|ID_S|\}k1(S,R), \{|TS_S|\}k1(S,R), m)), send_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1),$
$read_3(s, r, \{|TS_{S'}|\}k3(S,R), \{|TS_S|\}k3(S,R), \{|k2|\}k3(S,R), m2), send_4(r, s, Ksession, k4(s,r), m3),$
$claim_8(r, secret, TS_{ER}), claim_9(r, secret, Ksession), claim_{10}(r, nisynch))$

PROOF OF SECRECY FOR EDGE ROUTER ID (ID_{ER})

Assume α is a trace with index r1 $\alpha_{r1} = (\theta_{r1}, \rho_{r1}, \sigma_{r1}), claim_5(s, secret, ID_{ER})$. Assume that the intruder learns tss we are going to derive a contradiction. Let k be the smallest index,
$\Rightarrow \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) \in M_{k+1}$
$\Rightarrow \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) not \in M_{k+1}$
According to the derivation rules, this increase in knowledge is because of send rule and deflect rule. smallest index p<k,
$\Rightarrow \alpha_p = (\theta^{'}, \rho^{'}, \sigma^{'}), send_l(m))$
$\alpha_p \Longrightarrow \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) \sqsubseteq \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(m)$
Since we have four possible send events in LAUP protocol, We have 4 cases: l=1,2,3,4
l=1: $\alpha_p = (\theta^{'}, \rho^{'}, \sigma^{'}), send_1(s, r, \{|ID_S|\}k1(s,r), \{|TS_S|\}, m)$ since ID_S and TS_S both differ from ID_{ER}, the intruder can not learn
$\alpha_p = \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$ from $\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(s, r, \{|ID_S|\}k1(s,r), \{|TS_S|\}, m)$ which yields contradiction.
l=2: $\alpha_p = (\theta^{'}, \rho^{'}, \sigma^{'}), send_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1)$
The intruder can learn ID_{ER} because $\rho^{'}(i)$ is an untrusted agent and either,

$$\langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) = \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{ER}) \tag{1}$$

or

$$\langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) = \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(ID_{ER}) \tag{2}$$

From equation 1,$\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{ER}) not \in M_p$, applying Lemmas 3.26 and 3.27 found in (21) to find s1 with
$\alpha_{s1} = (\theta_{s1}, \rho_{s1}, \sigma_{s1}), send_1(s, r, \{|ID_S|\}k1(S,R), \{|TS_S|\}, m)$
This gives $\langle\theta_{s1}, \rho_{s1}, \sigma_{s1}\rangle(TS_S) = \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{ER}) = \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$ which cannot be the case, since TS_S and ID_{ER} are distinct constants. From equation 2, using Lemma 3.28 of (21)we derive $\theta_{r1} = \theta^{'}$since run identifiers are unique we have $\rho_{r1} = \rho^{'}$
so $\rho_{r1}(i) = \rho^{'}(i)$ which contradict the assumption that $\rho_{r1}(i)$ is a trusted agent.
l=3: $\alpha_p = ((\theta^{'}, \rho^{'}, \sigma^{'}), send_3(s, r, \{|TS_{S'}|\}k3(s,r), \{|TS_S|\}, k3(s,r), \{|k2|\}k3(s,r)m2)$ in order to learn $\langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$ from $\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle), send_3(s, r, \{|TS_{S'}|\}k3(s,r), \{|TS_S|\}, k3(s,r), \{|k2|\}k3(s,r)m2)$ we must have $\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{S'}) = \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$ and $\rho^{'}(r)$ is an untrusted agent.
Using Lemma 3.26 of (21), we can find index i2,
$\alpha_{i2} = (\theta^{'}, \rho^{'}, \sigma^{'}, read_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}, k2(s,r), m1)$ because $\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{S'}) not \in M_p$ we can aply lemma 3.27 of (21) to find index r2 with
$\alpha_{r2} = ((\theta_{r2}, \rho_{r2}, \sigma_{r2}), send_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1))$
This gives $\rho^{'}(r) = \rho_{r2}(r)$ next we derive $\langle\theta_{r2}, \rho_{r2}, \sigma_{r2}\rangle(ID_{ER}) = \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{S'}) = \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$. Applying Lemma 3.28 of (21) yields,
$\theta_{r2} = \theta_{r1}$ and thus $\rho_{r2} = \rho_{r1}$ so, $\rho^{'}(r) = \rho_{r2}(r) = \rho_{r1}(r)$

because $\rho^{'}(r)$ is an untrusted agent, while $\rho_{r1}(r)$ is trusted. We obtain contradiction. Similarly l=4 case will be proved to obtain contradiction.

This finishes the proof of claim, secrecy of ID_{ER}.

PROOF OF NON-INJECTIVE SYNCHRONIZATION:

Let $\alpha \in Trace(LAUP)$, for some r9 and $(\theta_r, \rho_r, \sigma_{r9}) \in Inst$, with $tme(\rho_r) \subseteq Agent_T$, we have $\alpha_{r9} = ((\theta_r, \rho_r, \sigma_{r9}), claim_{10}(r, nisynch))$. We are going to find a run executing the initiator role which synchronises on the events labeled 1, 2 and 3, since prec(LAUP,9)=1,2,3. By Lemma 3.26 found in (21), we find $r1, r2, r3 (r1 < r2 < r3 < r9)$ and $\sigma_{r1} \subseteq \sigma_{r2} \subseteq \sigma_{r3} \subseteq \sigma_{r9}$, such that $\alpha_{r1} = ((\theta_r, \rho_r, \sigma_{r1}), read_1(s, r, \{|ID_S|\}k1(S,R), \{|TS_S|\}k1(S,R), m))$

$\alpha_{r2} = ((\theta_r, \rho_r, \sigma_{r2}), send_2(r, s, \{|ID_{ER}|\}k2(S,R), \{|TS_{ER}|\}k2(S,R), m1)$

$\alpha_{r3} = ((\theta_r, \rho_r, \sigma_{r3}), read_3(s, r, \{|TS_{S'}|\}k3(S,R), \{|TS_S|\}k3(S,R), \{|k2|\}k3(S,R), m2)$.

We have proved that ider remains secret, so we can apply Lemma 3.27 found in (21) and find index s3 and $(\theta_s, \rho_s, \sigma_{s3})$ such that s3 < r3 and

$\alpha_{s3} = ((\theta_s, \rho_s, \sigma_{s3}), send_3(s, r, \{|ntss|\}k3(s,r), \{|TS_S|\}k3(s,r),$
$\{|k2|\}k3(s,r), m2)) \bigwedge \langle \theta_r, \rho_r, \sigma_{r3} \rangle (ID_{ER}) = (\theta_s, \rho_s, \sigma_{s3}(TS_{S'}))$. Applying Lemma 3.26 found in (21) we obtain $s_1 < s_2 < s_3$ such that

$\alpha_{s1} = ((\theta_s, \rho_s, \sigma_{s1}), send_1(s, r\{|ID_S|\}k1(s,r), \{|TS_S|\}k1(s,r), m))$

$\alpha_{s2} = ((\theta_s, \rho_s, \sigma_{s2}), read_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1))$

$\alpha_{s3} = ((\theta_s, \rho_s, \sigma_{s3}), send_3(s, r, \{|TSS_{S'}|\}k3(s,r), \{|TS_S|\}, k3(s,r), \{|k2|\}k3(s,r), m2))$.

We found that θ_s is a candidate, we need to prove that it synchronizes with run θ_r. Therefore we are going to establish r2 < s2, s1 < r1 and that the corresponding send and read events match each other.

Observing α_{s2}, Since $\langle \theta_r, \rho_r, \sigma_{r3} \rangle (ID_{ER})$ is secret, $\langle \theta_s, \rho_s, \sigma_{s2} \rangle (TS_{S'})$ is secret too and we can apply Lemma 3.27 of (21), obtaining index $r2^{'} < s2$ such that

$\alpha_{r2'} = ((\theta_{r'}, \rho_{r'}, \sigma_{r2'}, send_2(r, s, \{|ID_{ER}|\}k2(S,R), \{|TS_{ER}|\}k2(S,R), m1))$ such that we have

$\langle \theta_s, \rho_s, \sigma_{s2} \rangle (\{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1) = \langle \theta_{r'}, \rho_{r'}, \sigma_{r2'} \rangle (\{|ID_{ER}|\}k2(S,R), \{|TS_{ER}|\}k2(S,R), m1)$. This implies that we have

$\langle \theta_r, \rho_r, \sigma_{r3} \rangle (ID_{ER}) = (\theta_s, \rho_s, \sigma_{s3}(TS_{S'})) = \langle \theta_{r'}, \rho_{r'}, \sigma_{r2'} \rangle (ID_{ER})$, so from Lemma 3.28 we have $\theta_r = \theta_{r'}$ and thus $r2 = r2^{'}$.

This establishes synchronization of events α_{s2} and α_{r2}.

Considering α_{r1}. Because$\langle \theta_r, \rho_r, \sigma_{r1} \rangle (ID_{ER})$ is secret, we can apply Lemma 3.27 of (21), which gives index $s1^{'} < r1$ such that

$\alpha_{s1'} = ((\theta_{s'}, \rho_{s'}, \sigma_{s1'}), send_1(s, r, \{|ID_S|\}k1(s,r), \{|TS_S|\}k1(s,r), m))$

$and \langle \theta_r, \rho_r, \sigma_{r1} \rangle (\{|ID_S|\}k1(S,R), \{|TS_S|\}k1(S,R), m)) = \langle \theta_{s'}, \rho_{s'}, \sigma_{s1'} \rangle (\{|ID_S|1(s,r), \{|TS_S|\}k1(s,r), m))$.

Correspondence of α_{s2} and α_{r2} gives,

$\langle \theta_s, \rho_s, \sigma_{s2} \rangle (TS_S) = \langle \theta_r, \rho_r, \sigma_{r2} \rangle (ID_{ER}) = \langle \theta_r, \rho_r, \sigma_{r1} \rangle (ID_{ER}) = \langle \theta_{s'}, \rho_{s'}, \sigma_{s1'} \rangle (TS_S)$.

By Lemma 3.28 θ_s and $\theta_{s'}$ are equal, which establishes synchronicity of events α_{r1} and α_{s1}.

This finishes the proof of Non-Injective synchronisation property of LAUP algorithm.

5 | SECURITY ANALYSIS BASED ON THREAT SCENARIOS:

The session key establishment and authentication method followed by LAUP algorithm are well suited for LoWPAN wireless network sensors. Because, the LAUP algorithm uses lightweight symmetric cryptographic methods to establish a session key and authentication process. Since the MAC address of the 6LoWPAN sensor device and Edge Router are in the encrypted form during the process of LAUP, it will not be disclosed to an eavesdropper. The proposed LAUP algorithm gives reliable protection against the well known LoWPAN security attacks.

REPLAY ATTACK: LAUP protects the transmission of messages from replay attack in all the four flights by adding MAC values, thereby the integrity of the message is maintained throughout the algorithm. So insertion, deletion or modification of messages could not be performed by the attacker. All the four flight information analyzed step by step for what would happen if the attacker captures the flight information. The first flight message packet could not be reproduced by the attacker because the packet contains information such as sensor$'s$ unique MAC ID and the timer value of sensor at the time of first flight message generation and most importantly they are appended with MAC_{one} value. The second flight message contains the nonce value of Edge Router encrypted with the unique $flight_{two}$key SK2. Also, we proved the secrecy of Edge Router ID with Scyther tool, so that adversaries cannot get this information and reproduce it.

This strongly encrypted value cannot be deciphered by the attacker since he does not know the nonce value of Edge Router. The third flight message contains nonce values used in the first flight and the nonce value of the third flight, encrypted with unique $flight_{three}$key SK3. These ciphertexts are cryptographically strong enough for the lightweight communication and cannot be replayed so that the integrity of the message maintained. The fourth flight message has MAC_{four} value and ciphered form of the session key. An attacker can get the session key only if he knows unique $flight_{four}$ key SK4. On the whole, nonce values and MAC values prevent the attacker from replaying the message and maintaining integrity.

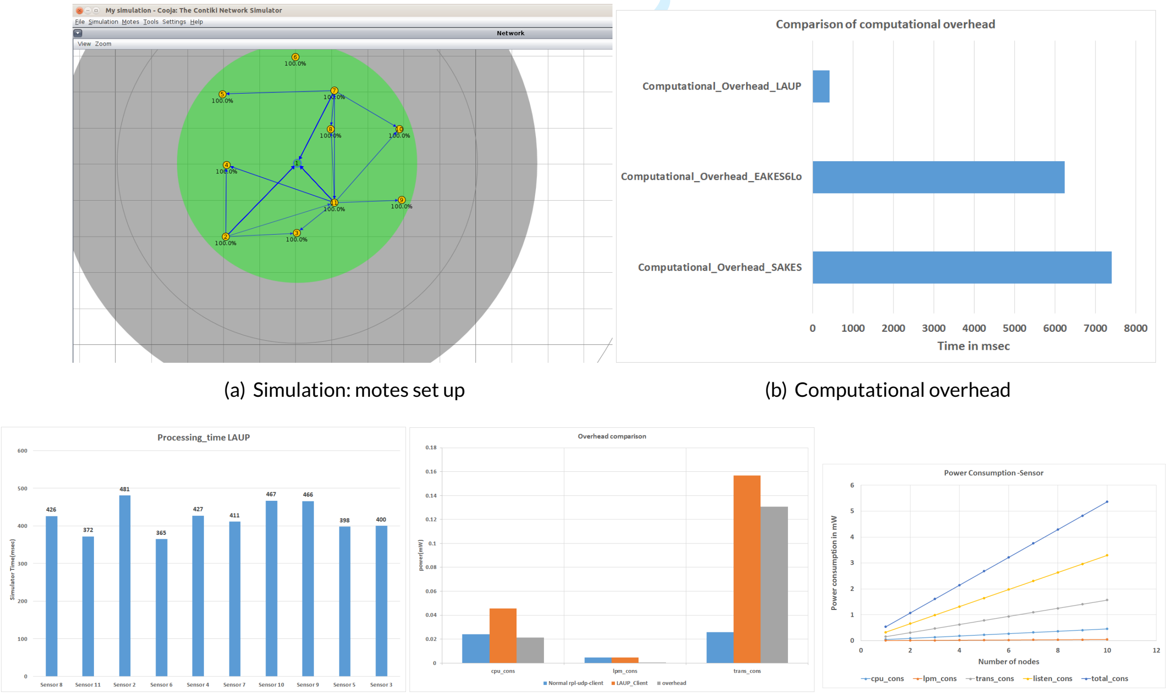
MAN IN THE MIDDLE ATTACK: LAUP protects the communication of messages against Man in the Middle attack. The man in the middle attacker possibly alters the communication between the two parties who believe that they are directly communicating with each other. But LAUP messages, in all four flights, are encrypted with the secure AES-128-ECB algorithm and unique flight keys.The flight messages are constructed with nonce values. Also, Non-Injective synchronization property is maintained.

IMPERSONATION ATTACK: Here an adversary can pretend like one of the legitimate sensors in the LoWPAN network. LAUP assumes all the sensors$'$ identity are registered with the Edge Router. Sensor hello request from an impersonation adversary rejected by checking its identity.

6 | VALIDATION AND EVALUATION OF PROPOSED LAUP ALGORITHM

6.1 | Evaluation using COOJA simulator

Our proposed LAUP algorithm for authentication and key distribution algorithm simulated in Contiki OS COOJA simulator environment. Our simulated environmental architecture is shown in Figure 4(a). We have taken Wismote as a sensor and the Edge Router as well. The scalability of our proposed LAUP algorithm is checked by adding 65 nodes to the network with the Edge Router in the COOJA simulator. LAUP simulated like an RPL (Routing Protocol for Lossy networks) UDP client-server application whereas an Edge Router acts as a server.



(a) Simulation: motes set up

(b) Computational overhead

(c) Processing Time

(d) Power consumption

(e) Power consumption based on no.of nodes

FIGURE 4 The COOJA simulator - Evaluation results of LAUP.

Wismote voltage value and various current values such as CPU current value, low power mode current value, transmission, and current reception values are taken from the Wismote datasheet (24). Power consumption is calculated using the formula found in (25). The sensor who wants to communicate with the Edge Router is consuming 0.0456 mw of CPU power, 0.0048 mw of low power mode (lpm) power, 0.1567 mw of transmission power, 0.3300 mw of reception power and 0.5371 mw of total power. We simulated our algorithm with ten sensors. The graph in Figure 4(b) explains the comparison of computational overhead of LAUP with EAKES6Lo and SAKES (15) overhead values given in (9).

TABLE 3 Memory usage of the sensor and an Edge Router

text	data	bss	dec	hex	filename
45619	350	13236	59205	e745	udp-clientv1.wismote
49253	402	13782	63437	f7cd	border-routerv2.wismote

Although we compared the authentication algorithms (EAKES6Lo, SAKES) which simulated in different environments, the LAUP authentication algorithm provides 15 times less computational overhead than EAKES6lo and 18 times less computational overhead than SAKES authentication algorithm for LoWPAN devices. Figure 4(e) shows the power consumption value increases as the number of nodes increases also Figure 4(d) graph tells us power consumption while receiving messages is high compared to the lpm, and transmission energy consumption in LoWPAN devices.

The difference in power consumption of conventional sensor (without any authentication) and LAUP sensor (with the proposed authentication algorithm) is explained in Figure 4(d), and LAUP consumes 0.13079624 mw more power while transmitting messages than the regular sensor communication without authentication. Each flight of LAUP is communicated as a payload of the transport layer. Flight 1 (SensorHello) and Flight 2 (ERConf) consume 64 bytes each. Flight 3 (SensorChallenge) consumes 80 bytes. Flight 4 (Finished) consumes 48 bytes. Graph in Figure 4(e) reveals the total processing time of the LAUP algorithm for different sensors over time. From this graph, taking the average of the total processing time of 10 sensors, we proved that our proposed LAUP algorithm takes less time to execute the full authentication algorithm. Up to 65 6LoWPAN devices can be connected without resource-exhaustion to the Edge Router in a specific position. Coding of LAUP will be sent to the reader upon request.

Memory usage of LAUP algorithm is calculated on the sensor, and the Edge Router based on the information found in (26). Table 3 summarizes the memory usage of the sensor and the Edge Router. Data segment refers to read-write data, and bss segment indicates zero-initialized data. The sum of text, data and bss values mentioned in dec section. Flash consumption of LAUP algorithm is 45969 bytes in sensor and 49655 bytes in the Edge Router. RAM use of LAUP algorithm is 13586 bytes in sensor and 14184 bytes in the Edge Router. The total processing time of our proposed LAUP algorithm showed in Figure 4(c) takes 421.3 msec which is comparatively lower than the processing time of existing algorithms such as EAKES6Lo and SAKES are given in (9). This total processing time of LAUP is calculated form the COOJA simulators' mote output window.

6.2 | Hardware Evaluation

Figure 5 shows the testbed setup of the LAUP evaluation. We use our laptop with Contiki OS installed on VMware workstation 12 player, two wismotes from Arago systems, one MSP430 USB-Debug-Interface(MSP-FET430UIF) to upload the contiki program into wismote hardware and one TI CC2531 Dongle to capture 6LoWPAN packets. Wismote operates on 2.4GHz free band and supports IPV6. Sensors/actuators such as temperature, humidity, light or 3D accelerometer are available on WiSMote.

To upload our authentication program into wismote hardware, we have installed msp430flasher Linux version in Contiki OS. While uploading, MSPFlasher utility of Contiki OS invokes MSP430 USB-Debug-Interface(MSP-FET430UIF) to flash the Wismote memory. Once the Border router started, it prefixes aaaa to the link local address of the 6lowpan device which resides within the coverage of it and sets the IPv6 address. The 6LoWPAN device which needs to get authenticated by the Border Router has the same PAN ID as the Border router. The wismote sensor who wants to communicate with the Edge Router is consuming 0.0214 mw of CPU power, 0.0080 mw of low power mode (lpm) power and 0.0088 mw of transmission power of total power. A comparative power consumption of LAUP on wismote using the simulator and a hardware implementation is shown in Figure 6(a). Power consumption values such as CPU consumption, lpm consumption and transmission consumption are compared between simulation results and hardware results in Figure 6(a). LAUP authentication process running on wismote hardware takes 35 ticks or 1.0681 milliseconds to get connected with the border router. The same LAUP process running on simulated wismote takes 51 ticks or 1.5563 milliseconds which is comparatively less than the simulation result. Figure 6(b) and 6(c) clearly shows the difference between LAUP total processing time using simulated wismote using COOJA simulator and a testbed evaluation methods.

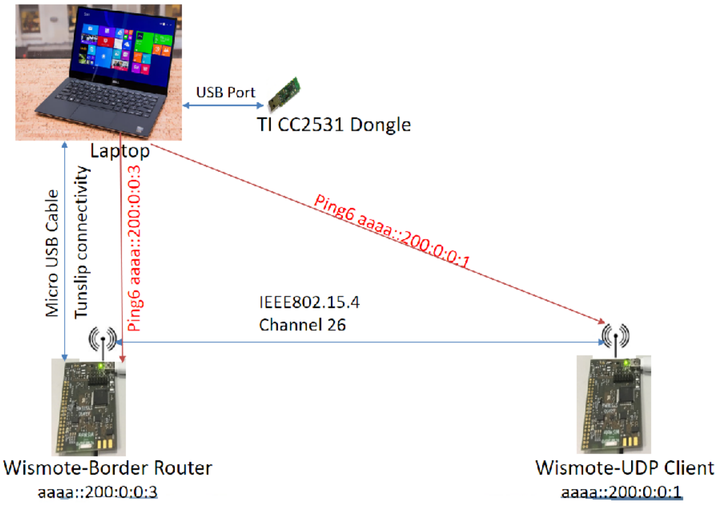
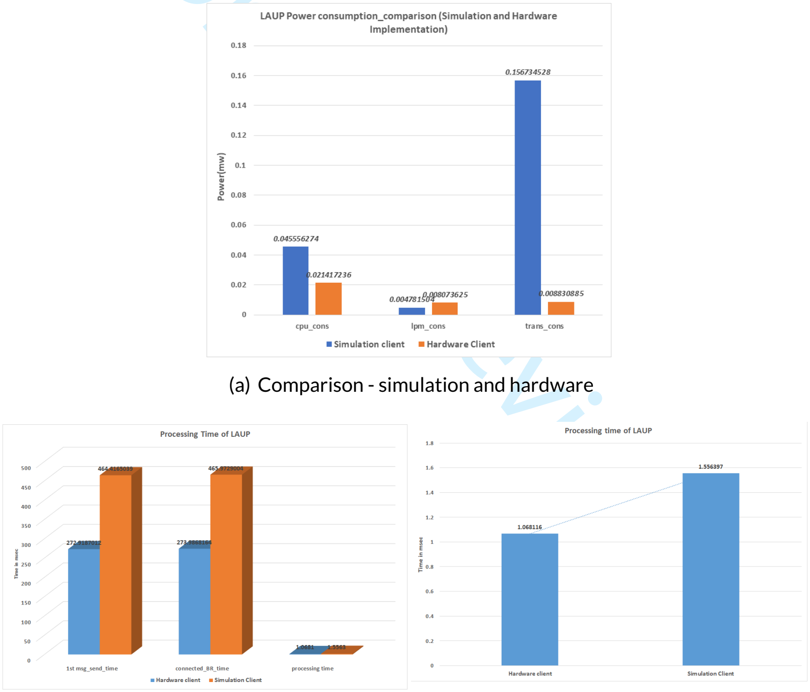


FIGURE 5 Hardware setup for LAUP



(a) Comparison - simulation and hardware

(b) Processing time of LAUP

(c) Processing time of LAUP

FIGURE 6 Test bed results on power consumption, and processing time of LAUP

7 | CONCLUSION

With the knowledge of existing algorithms and their limitations in the field of authentication and key distribution, we proposed our LAUP algorithm to overcome these limitations. Our algorithm formally verified by the formal verification tool called "Scyther", and we proved that authentication properties such as Aliveness, NonInjective Agreement, Secrecy of keys and Non-Injective Synchronization are maintained. Moreover, LAUP algorithm works with UDP protocol and possible threats such as Replay attack, Man in the Middle attack and impersonation attack are analyzed theoretically in section 5. In addition to the formal verification proof, we presented simulation results using the Contiki OS COOJA simulator with ten 6LoWPAN sensors as clients and one Edge Router. The hardware evaluation results broadly supported our simulation results and theoretical predictions. Evaluation of the proposed algorithm carried out based on the clock ticks of wismote. From our evaluation results, we can say that our algorithm is highly secured since LAUP generates the respective keys for each flight using the nonce value of sensors and Edge Router. Additionally, this LAUP algorithm is flexible to update the keys after each session. In future, LAUP will be tested against various attacks such as Sybil attacks, DoS attacks and replay attacks using Cooja simulator and the hardware. From the verification tool results, evaluation results from the simulator and hardware, we proved that the LAUP algorithm for authentication and key distribution is faster, highly secured, scalable for LoWPAN networks and flexible enough to update the keys dynamically.

ACKNOWLEDGMENTS

We would like to thank NICTA and DAT61 for their support and funding for this project, Project no: 1032350.

References

[1] Shelby Zach, Bormann Carsten. *6LoWPAN: The wireless embedded Internet*. John Wiley & Sons; 2011.

[2] Mahmoud Rwan, Yousuf Tasneem, Aloul Fadi, Zualkernan Imran. Internet of things (IoT) security: Current status, challenges and prospective measures. In: :336-341IEEE; 2015.

[3] Al-Fuqaha Ala, Guizani Mohsen, Mohammadi Mehdi, Aledhari Mohammed, Ayyash Moussa. Internet of things: A survey on enabling technologies, protocols, and applications. *Communications Surveys & Tutorials, IEEE*. 2015;17(4):2347-2376.

[4] Sheng Zhengguo, Yang Shusen, Yu Yifan, Vasilakos Athanasios, Mccann Julie, Leung Kin. A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wireless Communications*. 2013;20(6):91–98.

[5] Gheorghe Laura, Rughinis Razvan, Deaconescu Razvan, Tapus Nicolae. Authentication and anti-replay security protocol for wireless sensor networks. In: :7–13IEEE; 2010.

[6] Jara Antonio J, Marin Leandro, Skarmeta Antonio FG, Singh Dhananjay, Bakul Gohel, Kim Daeyeoul. Mobility modeling and security validation of a mobility management scheme based on ECC for IP-Based Wireless Sensor Networks (6LoWPAN). In: :491-496IEEE; 2011.

[7] Al Fardan Nadhem J, Paterson Kenneth G. Lucky thirteen: Breaking the TLS and DTLS record protocols. In: :526-540IEEE; 2013.

[8] Raza Shahid, Voigt Thiemo, Jutvik Vilhelm. Lightweight IKEv2: a key management solution for both the compressed IPsec and the IEEE 802.15. 4 security. In: Citeseer; 2012.

[9] Qiu Yue, Ma Maode. A Mutual Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN Networks. *IEEE Transactions on Industrial Informatics*. 2016;12(6):2074–2085.

[10] Esfahani Alireza, Mantas Georgios, Matischek Rainer, et al. A lightweight authentication mechanism for M2M communications in industrial IoT environment. *IEEE Internet of Things Journal*. 2017;.

[11] Raza Shahid, Trabalza Daniele, Voigt Thiemo. 6LoWPAN compressed DTLS for CoAP. In: :287–289IEEE; 2012.

[12] Krentz Konrad-Felix, Rafiee Hosnieh, Meinel Christoph. 6LoWPAN security: adding compromise resilience to the 802.15. 4 security sublayer. In: :1ACM; 2013.

[13] Perrig Adrian, Szewczyk Robert, Tygar Justin Douglas, Wen Victor, Culler David E. SPINS: Security protocols for sensor networks. *Wireless networks*. 2002;8(5):521–534.

[14] Krentz Konrad-Felix, Meinel Christoph. Handling reboots and mobility in 802.15. 4 security. In: :121–130ACM; 2015.

[15] Hussen Hassen Redwan, Tizazu Gebere Akele, Ting Miao, Lee Taekkyeun, Choi Youngjun, Kim Ki-Hyung. SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6L0WPAN). In: :246–251IEEE; 2013.

[16] Li Ming, Yu Shucheng, Guttman Joshua D, Lou Wenjing, Ren Kui. Secure ad hoc trust initialization and key management in wireless body area networks. *ACM Transactions on sensor Networks (TOSN)*. 2013;9(2):18.

[17] Pietro Roberto Di, Ma Di, Soriente Claudio, Tsudik Gene. Self-healing in unattended wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*. 2012;9(1):7.

[18] Premnath Sriram N, Haas Zygmunt J. Security and Privacy in the Internet-of-Things Under Time-and-Budget-Limited Adversary Model. *Wireless Communications Letters, IEEE*. 2015;4(3):277-280.

[19] Hennebert Christine, Dos Santos Jessye. Security protocols and privacy issues into 6lowpan stack: A synthesis. *Internet of Things Journal, IEEE.* 2014;1(5):384-398.

[20] Mukherjee Amitav. Physical-Layer Security in the Internet of Things: Sensing and Communication Confidentiality Under Resource Constraints. *Proceedings of the IEEE.* 2015;103(10):1747-1761.

[21] Cremers Casimier Joseph Franciscus. *Scyther: Semantics and verification of security protocols.* Eindhoven University of Technology; 2006.

[22] Lowe Gavin. A hierarchy of authentication specifications. In: :31–43IEEE; 1997.

[23] Cremers Cas, Mauw Sjouke. Operational semantics of security protocols. In: Springer 2005 (pp. 66–89).

[24] Instruments Texas. *CC2520 DATASHEET, 2007.* 2014.

[25] SonHan . Thigschat Internet of things http://thingschat.blogspot.com.au/2015/04/contiki-os-using-powertrace-and.html/[Online; accessed 12-November-2016]; 2016.

[26] Velinov Aleksandar, Mileva Aleksandra. Running and Testing Applications for Contiki OS Using Cooja Simulator. 2016;.

AUTHOR BIOGRAPHY

Annie Gilda Roselin received **ME** degree in Computer Science and Engineering from Mepco Schlenk Engineering College, Sivakasi, Anna University, Chennai and **B.Tech** degree in Information Technology from Noorul Islam College of Engineering, Anna University, Chennai. She has worked as a Lecturer in Computer Science and Engineering department of Velammal Engineering College, Chennai, India for three years from 2008 to 2011. She is currently doing Ph.D. degree in 6LoWPAN protocol security from School of Electrical and Data Engineering, University of Technology, Sydney.

Dr.Priyadarsi Nanda is a Senior Lecturer at the University of Technology Sydney (UTS) with extensive experience in research and development of Cyber Security, IoT security, and wireless sensor network security. His most significant work has been in the area of Intrusion Detection and Prevention systems using image processing techniques, Sybil attack detection in IoT based applications, intelligent firewall design. He has published 80+ high quality refereed research papers including Transactions on Computers, Transactions in Parallel Processing and Distributed Systems, Future Generations of Computer Systems (FGCS). He has successfully supervised 10 research students in the past and currently supervising 7 research students in Cybersecurity research. Dr. Nanda holds a PhD in Computing Science, Master in Computer Engineering and Bachelor in Computer Engineering.

Dr.Surya Nepal is a Principal Research Scientist at Information Engineering Laboratory of CSIRO Computational Informatics. Dr. Nepal has 15+ years experience in computer science research, latterly with a specific focus on security, privacy and trust in distributed systems. He has more than 100 publications to his credit, has edited or co-authored several books, and is the co-inventor of two patents. Much of his work appears in top international forums such as VLDB, ICDE, ICWS, SCC, CoopIS, ICSOC, International Journals of Web Services Research, IEEE Transactions on Service Computing, IEEE TPDS, ACM Computing Survey and ACM Transaction on Internet Technology. He obtained his BE from the National Institute of Technology, Surat, India; his ME from the Asian Institute of Technology, Bangkok, Thailand; and Ph.D. from RMIT University, Australia.

Dr.Sean He as a Chief Investigator, has received various research grants including four national Research Grants awarded by Australian Research Council (ARC). He is the Director of Computer Vision and Pattern Recognition Laboratory at the Global Big Data Technologies Centre (GBDTC) at the University of Technology Sydney (UTS). He is an IEEE Senior Member and has been an IEEE Signal Processing Society Student Committee member. He has been awarded 'Internationally Registered Technology Specialist' by International Technology Institute (ITI). He is a leading researcher in several research areas including big-learning based human behaviour recognition on a single image, image processing based on hexagonal structure, authorship identification of a document and a document's components (e.g., sentences, sections etc.), network intrusion detection using computer vision techniques, car license plate recognition of high speed moving vehicles with changeable and complex background, and video tracking with motion blur.

□

Received: 00 Month 0000 | Revised: 00 Month 0000 | Accepted: 00 Month 0000
DOI: xxx/xxxx

ARTICLE TYPE

Testbed Evaluation of Light weight Authentication Protocol(LAUP) for 6LoWPAN wireless sensor networks

Annie Gilda Roselin[1,2] | Priyadarsi Nanda[1] | Surya Nepal[2] | Sean He[1]

[1]School of Electrical and Data Engineering,
University of Technology Sydney(UTS),
Sydney, Australia
[2]CSIRO, Data61,Marsfield, NSW, Australia

Correspondence
Annie Gilda Roselin, Email:
Annie.G.ArockiaBaskaran@student.uts.edu.au

Summary

6LoWPAN networks involving wireless sensors consist of resource starving miniature sensor nodes. Since secured authentication is one of the important considerations, use of asymmetric key distribution scheme may not be a perfect choice. Recent research shows that Lucky Thirteen attack has compromised Datagram Transport Layer Security (DTLS) with Cipher Block Chaining (CBC) mode for key establishment. Even though EAKES6Lo and S3K techniques for key establishment follow the symmetric key establishment method, they strongly rely on a remote server and trust anchor. Our proposed Lightweight Authentication Protocol (LAUP) used a symmetric key method with no preshared keys and comprised of four flights to establish authentication and session key distribution between sensors and Edge Router in a 6LoWPAN environment. Each flight uses freshly derived keys from existing information such as PAN ID (Personal Area Network IDentification) and device identities. We formally verified our scheme using the Scyther security protocol verification tool. We simulated and evaluated the proposed LAUP protocol using COOJA simulator and achieved less computational time and low power consumption compared to existing authentication protocols such as the EAKES6Lo and SAKES. And LAUP is evaluated using real-time test bed and achieved less computational time which is supportive of our simulated results.

KEYWORDS:
6LoWPAN, Authentication, Session key

1 | INTRODUCTION

The network of low power sensors called LoWPAN (Low Power Wireless Personal Area Network) consists of small sensors with limited memory, less computational capability and low in resources. 6LoWPAN is one of the most important communication protocols used in LoWPAN network. 6LoWPAN is designed in such a way that it can transmit the IPv6 packets over LoWPAN network (1)(2)(3) (4). Security in LoWPAN is provided through authentication of sensor device before any communication happens.

Last MAC and Handshake Authentication methods are mainly used to provide anti-replay protection and authentication (5) for low power devices. The MAC value (hashed value of the message and key) of the previous message is appended to the current message, and the receiver validates the received MAC with the already stored MAC value. However, this technique assumes that the receiver will always accept the first message packet and globally shared keys for authentication. Even though ECC based Diffie-Hellman key exchange with Kerberos authentication (6) for wireless sensor networks provide high security, symmetric encryption and handshake authentication are the highly desirable mechanisms to build the authentication for the successive message transmissions of LoWPAN devices. The public key method of key distribution requires more computational time and energy which leads to overhead for lightweight sensors.

TABLE 1 Theoretical comparison and Features of LAUP

No	Parameters	Compressed DTLS	EAKES6Lo	S3K	LAUP
1	Asymmetric method	yes	No	No	No
2	Symmetric + preshared keys	No	Yes	Yes	No
3	Use of Key Distribution centre	No	Yes	Yes	No
4	Symmetric + No preshared keys	No	No	No	Yes

The plaintext of LoWPAN communication can be recovered from a DTLS connection using an OpenSSL implementation of DTLS when using CBC (Cipher Block Chaining) mode encryption (7). So it is not a good idea to have DTLS for Key management otherwise DTLS-CBC mode has to be enhanced to manage Lucky thirteen type attacks. To overcome these attacks, LAUP uses ECB (Electronic Code Book) mode of AES-128 encryption. Moreover, compressed DTLS has six flights for authentication, whereas our proposed LAUP has only four flights for authentication and session key distribution. Existing mechanisms such as lightweight IKEv2, EAKES6Lo, S3K and compressed DTLS are providing lightweight authentication for wireless sensor communication. Lightweight IKEv2 (8) is secure but requires more memory for calculation. EAKES6Lo (9) uses hash functions to ensure the integrity of messages. Though this technique uses symmetric key cryptography, it assumes that the secret key distributed to every node by the remote server. The distribution of secret keys remains a challenge.

S3K symmetric key establishment for IoT (8) uses trust anchor and a resource server for key distribution among clients. However, the secret key is shared between trust anchor and resource server before deployment. The limitation of this idea is that it is not scalable for a vast network and mobility nodes. Maintaining the uniqueness of shared key between trust anchor and resource server is complicated. S3K runs in addition to DTLS and CoAP protocols which in turn, increases the computational overhead of LoWPAN devices. Moreover, the secure connection has to be established between trust anchor and remote server before the key generation process using TLS/DTLS.

After a thorough study on key distribution and authentication of 6LoWPAN networks, we come to the following conclusions. The existing algorithms, which follow pre-shared key methods are facing problems while updating the keys and having the assumptions of pre-shared keys. Moreover, they are relying on a key distribution center (10) to distribute the keys which in turn cost more resources and providing protection to key distribution center are another problem in real time. Well established existing algorithms which are using asymmetric method for authentication and key distribution are having the limitation of spending more computational time for authentication. To overcome these existing constraints, we proposed our LAUP algorithm for authentication and key establishment. LAUP leads to the highly secured and easily adaptable authentication method for 6LoWPAN networks. Table 1 shows the explanation of our approach to 6LoWPAN networking.

Our proposed LAUP authentication algorithm addresses the significant challenges such as a distribution of preshared keys and usage of resource centers for key distribution in the field of authentication among low power devices. We made observations on how the conventional network protocols compressed to be compatible with LoWPAN Wireless Sensor Networks (WSN). Our observation showed that Compressed DTLS Handshake (11) uses six flights for authentication and key exchange, whereas our protocol uses only four flights for the same. To our knowledge, LAUP authentication algorithm is the most suitable for 6LoWPAN network and secure authentication algorithm without using preshared keys for authentication and key distribution of LoWPAN devices.

The rest of the paper is organized as follows. Section 2 explains related works in the area of authentication and key distribution of LoWPAN networks. Section 3 describes our proposed work using session request phase, authentication phase, and key distribution phase. Formal verification using Scyther tool is presented in section 4. Section 5 analyses the efficiency of the LAUP protocol against various attacks of LoWPAN network. Performance evaluation of LAUP using Contiki OS COOJA simulator and a real-time test bed is demonstrated in section 6. Finally, we conclude the paper in section 7.

2 | RELATED WORK

APKES (Adaptive Pairwise Key Establishment Scheme) (12) uses pre-distributed pairwise keys to derive pairwise session keys for authentication. SPINS (Security Protocol for Sensor Networks) (13) scheme provides security for wireless sensor communication also obtains keys from the pre-distributed master keys. Although AKES (Adaptive Key Establishment Scheme) (14) system uses PAN ID and address of the sensor for authentication and key distribution, it follows pre-distribution of keys to derive pairwise session keys. Unlike our LAUP, AKES uses the address of sensors to get the shared secret keys, whereas LAUP uses MAC ID of sensors. Moreover, sensors (LoWPAN devices) which are using AKES scheme for authentication, to be preloaded with any of the relevant addressing information like 8-byte extended, 2-byte short or 1-byte simple address. Since

TABLE 2 Key Derivation Process

Flight No	Key	Process
1	SK1	PANID
2	SK2	ID_S XOR *Nonce*$_S$ XOR SK1
3	SK3	ID_{ER} XOR SK2
4	SK4	*Nonce*$_{S'}$ XOR *Nonce*$_S$ XOR SK3
SESSIONKEY	$K_{Session}$	SK1 XOR SK2 XOR SK3 XOR SK4 XOR *Nonce*$_{ER'}$

LAUP uses MAC ID of devices, it does not need any reloading of address. APKES, SPINS and AKES methods discussed previously, but in most cases, attention directed towards pre-distributed keys for key distribution and authentication.

SAKES (15) and EAKES6Lo authentication schemes deal with pre-shared keys among the 6LoWPAN host, 6LoWPAN router, and 6LoWPAN edge router. EAKES6Lo (9) has three phases such as pre-deployment phase, authentication and key establishment phase and handover phase. The remote server distributes private/public keys used by the sensors during authentication. A registration request by the sensor node is sent to the server with sensor ID and its public key. This public key is derived using ECDH (Elliptic Curve Diffie-Helman) mechanism which is more power consuming process for LoWPAN devices. But LAUP eliminates this additional power requirement by not using a remote server for preshared keys. LAUP focuses on preventing attacks on transportation layer such as replay attack, a man in the middle attack, and impersonation attack.

GDP (Group Device Pairing) does not need extra hardware devices for preshared keys. Based on symmetric key cryptographic techniques GDP provides secure communication between wireless body area networks. Even though GDP method (16) supports no redistribution of keys, GDP needs human user intervention for verification during authentication. Periodical updates of local keys (17) could prevent sensor compromisation on static nodes. Smaller cryptographic keys play a significant role in providing security for sensor communication (18). Unlike GDP, LAUP does not need human intervention during authentication and key establishment process. Moreover, LAUP session keys are small and have a periodical update for each session.

3 | PROPOSED WORK

Our proposed LAUP provides security to the 6LoWPAN device communication by authenticating the intended 6LoWPAN devices with Edge Router. 6LoWPAN is designed in such a way that it can transmit the IPv6 packets over Low Power Personal Area Network. 6LoWPAN protocol stack adopts bottom-most two layers from IEEE 802.15.4. 6LoWPAN acts as an adaptation layer between the link layer and the network layer. Figure 1(a) shows 6LoWPAN protocol stack and examples of protocols used in each layer. IEEE 802.15.4 supports only 127-byte packet length of messages. But the Maximum Transferrable Unit (MTU) of IPV6 is 1280 bytes. 6LoWPAN adaptation layer provides fragmentation and re-ordering, and compression of the protocol stack headers of IPv6 packets to maintain the communication compatibility between IEEE 802.15.4 frame and the legacy Internet message packet (1), (19), (20). LAUP works on transport layer of the 6LoWPAN protocol.

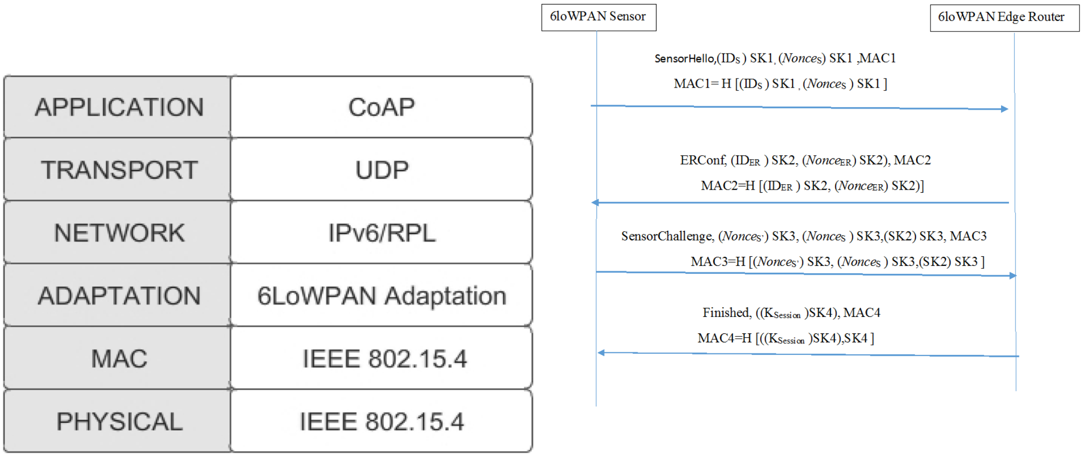
System architecture of LAUP is a hybrid of simple LoWPAN and an Ad-hoc LoWPAN architecture (1) shown in Figure 4(a). This architecture includes a 6LoWPAN device which is intended to communicate with the 6LoWPAN Edge Router and an Edge Router. LAUP works on the 6LoWPAN wireless sensor network communication between the 6LoWPAN device and the Edge Router. To maintain the secure communication 6LoWPAN device has to reside in the coverage range of Border Router.

3.1 | Basic assumptions of proposed work

Every sensor identity ID_S (MAC Address of sensor) is registered with the 6LoWPAN Edge Router ($6L_{ER}$) and they are physically secured. Our LAUP deals with 6LoWPAN devices which are deployed within the coverage range of 6LoWPAN Edge Router $6L_{ER}$. $6L_{ER}$ knows the PAN ID of LoWPAN network, and we assume that the sensors connected to the LoWPAN network are physically secured. We address the authentication and session key establishment of sensors when they are communicating to $6L_{ER}$. Each flight calculates its key for session key distribution by following common key derivation method. LAUP does not use any software or hardware based random number generation scheme to produce nonce values. Instead, the time of message generation on each flight has been taken as nonce respectively. By this way of receiving a nonce value, reduces the extra computational complexity and memory usage of low power devices. Table 2 explains the key derivation method to calculate unique flight keys. These methods use simple XOR functions with available values such as PAN ID, MAC ID and nonce values.

3.2 | Proposed LAUP Scheme

LAUP allows sensors of LoWPAN networks to communicate with a router to get cryptographically secure session key by two-level authentication using MAC ID of sensors and their nonce values. Figure 1(b) shows the process of LAUP communication between the 6LoWPAN sensor and an Edge Router.



APPLICATION	CoAP
TRANSPORT	UDP
NETWORK	IPv6/RPL
ADAPTATION	6LoWPAN Adaptation
MAC	IEEE 802.15.4
PHYSICAL	IEEE 802.15.4

6loWPAN Sensor
6loWPAN Edge Router
SensorHello,(ID_S) SK1, $(Nonce_S)$ SK1 ,MAC1
MAC1= H [(ID_S) SK1 , $(Nonce_S)$ SK1]
ERConf, (ID_{ER}) SK2, $(Nonce_{ER})$ SK2), MAC2
MAC2=H [(ID_{ER}) SK2, $(Nonce_{ER})$ SK2)]
SensorChallenge, $(Nonce_{S'})$ SK3, $(Nonce_S)$ SK3,(SK2) SK3, MAC3
MAC3=H [$(Nonce_{S'})$ SK3, $(Nonce_S)$ SK3,(SK2) SK3]
Finished, $((K_{Session})$SK4), MAC4
MAC4=H [$((K_{Session})$SK4),SK4]

(a) 6LoWPAN protocol stack (b) Proposed Authentication Algorithm

FIGURE 1 The 6LoWPAN protocol stack and the proposed LAUP algorithm.

LAUP algorithm gives protection against a Replay Attack, Man in the Middle attack, and impersonation attack by including the MAC values and nonce values. Even the attacker eavesdropped the message; he can not be able to reproduce the same message since the message is in the encrypted form and it needs the exact time of when the packet was generated. As a result of our LAUP algorithm, a unique session key will be produced by the Edge Router for a sensor claims **SensorHello** request. Figure 2 shows the flow of communication of LAUP on a 6LoWPAN device and the Edge Router. To encrypt the messages in each flight, we use the AES-128-ECB algorithm. A simple XOR function is used as a hash function to produce MAC values in all the four flights of communication. Our proposed LAUP algorithm has three phases (Session Request, Authentication and Key Distribution Phase) for authentication and session key establishment process.

3.2.1 | Session Request phase

Session request phase comprises of $flight_{one}$ communication message. In this phase, the 6LoWPAN sensor which is intended to communicate with the 6LoWPAN Edge Router $6L_{ER}$ sends the following content in its payload to the $6L_{ER}$. PAN ID of the network acts as a $flight_{one}$ key to encrypt the messages involved in first flight communication. The identity (MAC_ID) and the timer value (time generated by the sensor) of the sensor is encrypted by the $flight_{one}$ key called SK1. MAC_{one} value is calculated by applying the XOR function on the encrypted messages. Encrypted identity, encrypted nonce of the sensor, MAC_{one} value and "SensorHello" message are sent as a first flight information to $6L_{ER}$. Hence the identity and the nonce value of the sensor is retrieved by the $6L_{ER}$.

3.2.2 | Authentication phase

After receiving the $flight_{one}$ information from sensor, MAC_{one} value is calculated at Edge Router by hashing the received encrypted values. Before validating the authenticity of the sensor, the received MAC_{one} value is compared with the calculated MAC_{one} value. If both the values are same, protection against replay attack can be ensured, and the authentication process is going to be carried out by the Edge Router.

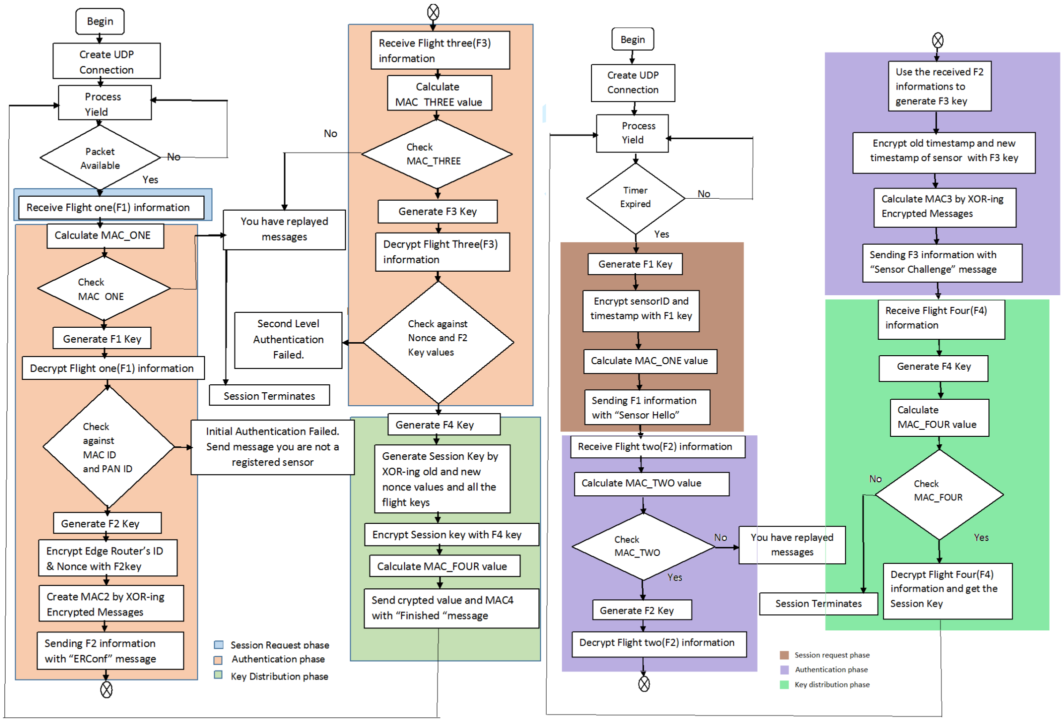
Two levels of authentication (Initial level and second level) will be performed by the $6L_{ER}$. Flow chart of Edge Router process in Figure 2 explains the two level authentication process in detail. In the Initial level of authentication, the received flight one information are decrypted using an AES-128 algorithm with ECB mode. Retrieved sensor MAC_ID from $flight_{one}$ information is checked against the already registered sensor MAC_ID. If a match is found, then $6L_{ER}$ generates $flight_{two}$ key SK2 by applying the XOR function on sensor MAC_ID, sensor nonce value and $flight_{one}$ key. Otherwise, 6LER send "You are not a registered sensor" message to the corresponding sensor and terminates the session.

$flight_{two}$ information is generated by the Edge Router and communicated with a sensor which claims authentication for session keys. Edge Router ID and nonce value is encrypted with $flight_{two}$ key SK2 and MAC_{two} is calculated by using a hash function on the resultant encrypted values. Along with the encrypted values and MAC_{two} values, "ERConf" message is sent to the sensor as a second flight information. After receiving $flight_{two}$ information, sensor checks the MAC_{two} value by calculating the same with the procedure followed by the Edge Router for MAC_{two} calculation. If the MAC_{two} is not replayed by any adversaries, then the sensor generates $flight_{two}$ key SK2 and decrypt the received $flight_{two}$ information from the Edge Router. After the above steps are performed, the sensor stores the ID of the Edge Router.

With the retrieved values from $flight_{two}$ packet, the sensor generates $flight_{three}$ key SK3 by applying XOR functions between Edge Router ID and $flight_{two}$ key SK2. $flight_{three}$ information are generated by encrypting the old and new nonce value of sensor and $flight_{two}$ key, with $flight_{three}$ key. Then MAC_{three} is calculated by applying a hash function on the encrypted values. Now $flight_{three}$ information packet is ready to send to Edge Router along with the "Sensor Challenge" message. When the Edge Router receives $flight_{three}$ information from sensor, Edge Router checks whether it has sent $flight_{two}$ information to the intended sensor.

Upon getting positive results of the checking operation, Edge Router starts to process the received $flight_{three}$ information from sensors. Initially Edge Router checks MAC_{three} value by calculating it and compare with the received MAC_{three} value. If the Edge Router found, the packet is not replayed, then proceed to calculate $flight_{three}$ key SK3 otherwise, terminates the session by sending "You have replayed the message".

$flight_{three}$ information are decrypted using $flight_{three}$ key SK3 and Edge Router gets the information like old, the new nonce value of sensor and $flight_{two}$ key SK2 calculated by the sensor. Edge Router does the second level authentication by comparing the nonce value of sensor what it has received from $flight_{two}$ and comparing $flight_{two}$ key SK2 value with the existing information. If the value matches, then it starts to process the further required session key generation. Thus the authentication phase of the sensor is completed by the Edge Router.


Begin
Create UDP Connection
Process Yield
Packet Available
No
Yes
Receive Flight one(F1) information
Calculate MAC_ONE
Check MAC ONE
Generate F1 Key
Decrypt Flight one(F1) information
Check against MAC ID and PAN ID
Generate F2 Key
Encrypt Edge Router's ID & Nonce with F2key
Create MAC2 by XOR-ing Encrypted Messages
Sending F2 information with "ERConf" message
You have replayed messages
Second Level Authentication Failed.
Session Terminates
Initial Authentication Failed. Send message you are not a registered sensor
Session Request phase
Authentication phase
Key Distribution phase
No
Receive Flight three(F3) information
Calculate MAC THREE value
Check MAC_THREE
Generate F3 Key
Decrypt Flight Three(F3) information
Check against Nonce and F2 Key values
Generate F4 Key
Generate Session Key by XOR-ing old and new nonce values and all the flight keys
Encrypt Session key with F4 key
Calculate MAC_FOUR value
Send crypted value and MAC4 with "Finished "message
Begin
Create UDP Connection
Process Yield
Timer Expired
No
Yes
Generate F1 Key
Encrypt sensorID and timestamp with F1 key
Calculate MAC_ONE value
Sending F1 information with "Sensor Hello"
Receive Flight two(F2) information
Calculate MAC_TWO value
Check MAC_TWO
No
Yes
You have replayed messages
Generate F2 Key
Decrypt Flight two(F2) information
Session Terminates
Session request phase
Authentication phase
Key distribution phase
Use the received F2 informations to generate F3 key
Encrypt old timestamp and new timestamp of sensor with F3 key
Calculate MAC3 by XOR-ing Encrypted Messages
Sending F3 information with "Sensor Challenge" message
Receive Flight Four(F4) information
Generate F4 Key
Calculate MAC_FOUR value
Check MAC_FOUR
No
Yes
Decrypt Flight Four(F4) information and get the Session Key

(a) Flow chart of Edge Router Process

(b) Flow chart of 6LoWPAN sensor Process

FIGURE 2 The process flow of LAUP on 6LoWPAN device and Edge Router.

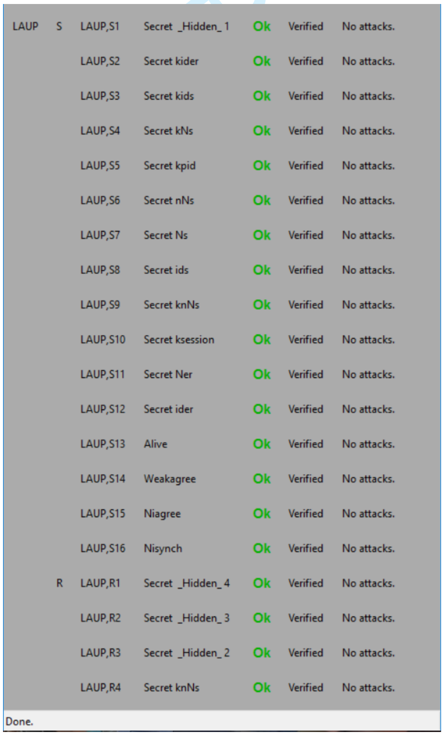
3.2.3 | Key Distribution phase

$flight_{four}$ key SK4 is the composition of old and new nonce values of the sensor and then the $flight_{three}$ key SK3. The session key ($K_{Session}$) is composed of $flight_{one}$ SK1, $flight_{two}$ SK2, $flight_{three}$ SK3, $flight_{four}$ SK4 and nonce of Edge Router at the time of session key generation. The session key is generated by the Edge Router by applying the XOR function on the above said values. The session key is encrypted with $flight_{four}$ key SK4 and is generated by the Edge Router. MAC_{four} value is calculated using XOR functions on encrypted values and $flight_{four}$ key SK4. After the cryptographic functions and MAC_{four} calculation, $flight_{four}$ information is sent to the intended sensor with the "Finished" message.

$flight_{four}$ information are received by the sensor and sensor generates $flight_{four}$ key SK4 using the same method followed by the Edge Router. A sensor checks whether the message is replayed or not by checking the MAC_{four} value with the calculated MAC_{four}. If the $flight_{four}$ message packet, through the checking operation of MAC values, then the sensor decrypts the $flight_{four}$ message and get the session key ($K_{Session}$). Thus the key distribution process is completely done by the Edge Router to the sensor by means of communicating four flight messages. This session key is used as a key to encrypt the further communication.

4 | FORMAL VERIFICATION OF LAUP ALGORITHM

The Scyther formal verification tool is used to verify the authentication properties of our proposed algorithm. Definitions of Aliveness, Secrecy, Non-Injective-Agreement, and Non-Injective-Synchronization are defined in (21, 22). Figure 3 shows that LAUP algorithm satisfies all the specified authentication properties such as aliveness, secrecy, non-injective-agreement, and non-injective-synchronization. We have proved the secrecy of Edge Router ID and Non-Injective Synchronization of LAUP based on (21, 23).



```
LAUP   S   LAUP,S1    Secret _Hidden_ 1   Ok   Verified   No attacks.
           LAUP,S2    Secret kider        Ok   Verified   No attacks.
           LAUP,S3    Secret kids         Ok   Verified   No attacks.
           LAUP,S4    Secret kNs          Ok   Verified   No attacks.
           LAUP,S5    Secret kpid         Ok   Verified   No attacks.
           LAUP,S6    Secret nNs          Ok   Verified   No attacks.
           LAUP,S7    Secret Ns           Ok   Verified   No attacks.
           LAUP,S8    Secret ids          Ok   Verified   No attacks.
           LAUP,S9    Secret knNs         Ok   Verified   No attacks.
           LAUP,S10   Secret ksession     Ok   Verified   No attacks.
           LAUP,S11   Secret Ner          Ok   Verified   No attacks.
           LAUP,S12   Secret ider         Ok   Verified   No attacks.
           LAUP,S13   Alive               Ok   Verified   No attacks.
           LAUP,S14   Weakagree           Ok   Verified   No attacks.
           LAUP,S15   Niagree             Ok   Verified   No attacks.
           LAUP,S16   Nisynch             Ok   Verified   No attacks.
       R   LAUP,R1    Secret _Hidden_ 4   Ok   Verified   No attacks.
           LAUP,R2    Secret _Hidden_ 3   Ok   Verified   No attacks.
           LAUP,R3    Secret _Hidden_ 2   Ok   Verified   No attacks.
           LAUP,R4    Secret knNs         Ok   Verified   No attacks.
Done.
```

FIGURE 3 Sycther tool results for LAUP Protocol verification

Secrecy: Secrecy expresses that certain information is not revealed to an intruder, even though we are communicating this data over an untrusted network. By maintaining the secrecy of edge router ID, we can perform second level authentication of 6LoWPAN devices with edge router also the intruder can not get any information of edge router ID.

Non-Injective Synchronization property of 6LoWPAN sensor, ensures that it communicates with the intended party 6LoWPAN Edge Router and the contents of receiving/sending messages are equal. Also, it guarantees the expected order of send and receives actions.

LAUP is specified as follows:

LAUP(s) $= \{s, r, (ID_{ER}, k1(s,r), k2(s,r), k3(s,r), k4(s,r)\},$
$send_1(s, r, \{|ID_S|\}k1(s,r), \{|TS_S|\}, m), read_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}, k2(s,r), m1),$
$send_3(s, r, \{|TS_{S'}|\}k3(s,r), \{|TS_S|\}, k3(s,r), \{|k2|\}k3(s,r)m2), read_4(r, s, Ksession, k4(s,r), m3),$
$claim_5(s, secret, ID_{ER}), claim_6(s, secret, TS_S), claim_7(s, nisynch))$
LAUP(r) $= (\{s, r, ID_S, TS_S, k1(s,r), k2(s,r), k3(s,r), k4(s,r)\},$
$read_1(s, r, \{|ID_S|\}k1(S,R), \{|TS_S|\}k1(S,R), m)), send_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1),$
$read_3(s, r, \{|TS_{S'}|\}k3(S,R), \{|TS_S|\}k3(S,R), \{|k2|\}k3(S,R), m2), send_4(r, s, Ksession, k4(s,r), m3),$
$claim_8(r, secret, TS_{ER}), claim_9(r, secret, Ksession), claim_{10}(r, nisynch))$

PROOF OF SECRECY FOR EDGE ROUTER ID (ID_{ER})

Assume α is a trace with index r1 $\alpha_{r1} = (\theta_{r1}, \rho_{r1}, \sigma_{r1}), claim_5(s, secret, ID_{ER})$. Assume that the intruder learns tss we are going to derive a contradiction. Let k be the smallest index,
$\Rightarrow \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) \in M_{k+1}$
$\Rightarrow \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) not \in M_{k+1}$
According to the derivation rules, this increase in knowledge is because of send rule and deflect rule. smallest index p<k,
$\Rightarrow \alpha_p = (\theta^{'}, \rho^{'}, \sigma^{'}), send_l(m))$
$\alpha_p \Longrightarrow \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) \sqsubseteq \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(m)$
Since we have four possible send events in LAUP protocol, We have 4 cases: l=1,2,3,4
l=1: $\alpha_p = (\theta^{'}, \rho^{'}, \sigma^{'}), send_1(s, r, \{|ID_S|\}k1(s,r), \{|TS_S|\}, m)$ since ID_S and TS_S both differ from ID_{ER}, the intruder can not learn
$\alpha_p = \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$ from $\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(s, r, \{|ID_S|\}k1(s,r), \{|TS_S|\}, m)$ which yields contradiction.
l=2: $\alpha_p = (\theta^{'}, \rho^{'}, \sigma^{'}), send_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1)$
The intruder can learn ID_{ER} because $\rho^{'}(i)$ is an untrusted agent and either,

$$\langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) = \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{ER}) \tag{1}$$

or

$$\langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER}) = \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(ID_{ER}) \tag{2}$$

From equation 1,$\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{ER}) not \in M_p$, applying Lemmas 3.26 and 3.27 found in (21) to find s1 with
$\alpha_{s1} = (\theta_{s1}, \rho_{s1}, \sigma_{s1}), send_1(s, r, \{|ID_S|\}k1(S,R), \{|TS_S|\}, m)$
This gives $\langle\theta_{s1}, \rho_{s1}, \sigma_{s1}\rangle(TS_S) = \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{ER}) = \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$ which cannot be the case, since TS_S and ID_{ER} are distinct constants. From equation 2, using Lemma 3.28 of (21)we derive $\theta_{r1} = \theta^{'}$since run identifiers are unique we have $\rho_{r1} = \rho^{'}$
so $\rho_{r1}(i) = \rho^{'}(i)$ which contradict the assumption that $\rho_{r1}(i)$ is a trusted agent.
l=3: $\alpha_p = ((\theta^{'}, \rho^{'}, \sigma^{'}), send_3(s, r, \{|TS_{S'}|\}k3(s,r), \{|TS_S|\}, k3(s,r), \{|k2|\}k3(s,r)m2)$ in order to learn $\langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$ from $\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle), send_3(s, r, \{|TS_{S'}|\}k3(s,r), \{|TS_S|\}, k3(s,r), \{|k2|\}k3(s,r)m2)$ we must have $\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{S'}) = \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$ and $\rho^{'}(r)$ is an untrusted agent.
Using Lemma 3.26 of (21), we can find index i2,
$\alpha_{i2} = (\theta^{'}, \rho^{'}, \sigma^{'}, read_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}, k2(s,r), m1)$ because $\langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{S'}) not \in M_p$ we can aply lemma 3.27 of (21) to find index r2 with
$\alpha_{r2} = ((\theta_{r2}, \rho_{r2}, \sigma_{r2}), send_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1))$
This gives $\rho^{'}(r) = \rho_{r2}(r)$ next we derive $\langle\theta_{r2}, \rho_{r2}, \sigma_{r2}\rangle(ID_{ER}) = \langle\theta^{'}, \rho^{'}, \sigma^{'}\rangle(TS_{S'}) = \langle\theta_{r1}, \rho_{r1}, \sigma_{r1}\rangle(ID_{ER})$. Applying Lemma 3.28 of (21) yields,
$\theta_{r2} = \theta_{r1}$ and thus $\rho_{r2} = \rho_{r1}$ so, $\rho^{'}(r) = \rho_{r2}(r) = \rho_{r1}(r)$

because $\rho^{'}(r)$ is an untrusted agent, while $\rho_{r1}(r)$ is trusted. We obtain contradiction. Similarly l=4 case will be proved to obtain contradiction.

This finishes the proof of claim, secrecy of ID_{ER}.

PROOF OF NON-INJECTIVE SYNCHRONIZATION:

Let $\alpha \in Trace(LAUP)$, for some r9 and $(\theta_r, \rho_r, \sigma_{r9}) \in Inst$, with $tme(\rho_r) \subseteq Agent_T$, we have $\alpha_{r9} = ((\theta_r, \rho_r, \sigma_{r9}), claim_{10}(r, nisynch))$. We are going to find a run executing the initiator role which synchronises on the events labeled 1, 2 and 3, since prec(LAUP,9)=1,2,3. By Lemma 3.26 found in (21), we find $r1, r2, r3 (r1 < r2 < r3 < r9)$ and $\sigma_{r1} \subseteq \sigma_{r2} \subseteq \sigma_{r3} \subseteq \sigma_{r9}$, such that $\alpha_{r1} = ((\theta_r, \rho_r, \sigma_{r1}), read_1(s, r, \{|ID_S|\}k1(S,R), \{|TS_S|\}k1(S,R), m))$
$\alpha_{r2} = ((\theta_r, \rho_r, \sigma_{r2}), send_2(r, s, \{|ID_{ER}|\}k2(S,R), \{|TS_{ER}|\}k2(S,R), m1)$
$\alpha_{r3} = ((\theta_r, \rho_r, \sigma_{r3}), read_3(s, r, \{|TS_{S'}|\}k3(S,R), \{|TS_S|\}k3(S,R), \{|k2|\}k3(S,R), m2)$.
We have proved that ider remains secret, so we can apply Lemma 3.27 found in (21) and find index s3 and $(\theta_s, \rho_s, \sigma_{s3})$ such that s3 < r3 and
$\alpha_{s3} = ((\theta_s, \rho_s, \sigma_{s3}), send_3(s, r, \{|ntss|\}k3(s,r), \{|TS_S|\}k3(s,r),$
$\{|k2|\}k3(s,r), m2)) \bigwedge \langle\theta_r, \rho_r, \sigma_{r3}\rangle(ID_{ER}) = (\theta_s, \rho_s, \sigma_{s3}(TS_{S'}))$. Applying Lemma 3.26 found in (21) we obtain $s_1 < s_2 < s_3$ such that
$\alpha_{s1} = ((\theta_s, \rho_s, \sigma_{s1}), send_1(s, r\{|ID_S|\}k1(s,r), \{|TS_S|\}k1(s,r), m))$
$\alpha_{s2} = ((\theta_s, \rho_s, \sigma_{s2}), read_2(r, s, \{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1))$
$\alpha_{s3} = ((\theta_s, \rho_s, \sigma_{s3}), send_3(s, r, \{|TSS_{S'}|\}k3(s,r), \{|TS_S|\}, k3(s,r), \{|k2|\}k3(s,r), m2))$.
We found that θ_s is a candidate, we need to prove that it synchronizes with run θ_r. Therefore we are going to establish r2 < s2, s1 < r1 and that the corresponding send and read events match each other.
Observing α_{s2}, Since $\langle\theta_r, \rho_r, \sigma_{r3}\rangle(ID_{ER})$ is secret, $\langle\theta_s, \rho_s, \sigma_{s2}\rangle(TS_{S'})$ is secret too and we can apply Lemma 3.27 of (21), obtaining index $r2^{'} < s2$ such that
$\alpha_{r2'} = ((\theta_{r'}, \rho_{r'}, \sigma_{r2'}, send_2(r, s, \{|ID_{ER}|\}k2(S,R), \{|TS_{ER}|\}k2(S,R), m1))$ such that we have
$\langle\theta_s, \rho_s, \sigma_{s2}\rangle(\{|ID_{ER}|\}k2(s,r), \{|TS_{ER}|\}k2(s,r), m1) = \langle\theta_{r'}, \rho_{r'}, \sigma_{r2'}\rangle(\{|ID_{ER}|\}k2(S,R), \{|TS_{ER}|\}k2(S,R), m1)$. This implies that we have
$\langle\theta_r, \rho_r, \sigma_{r3}\rangle(ID_{ER}) = (\theta_s, \rho_s, \sigma_{s3}(TS_{S'})) = \langle\theta_{r'}, \rho_{r'}, \sigma_{r2'}\rangle(ID_{ER})$, so from Lemma 3.28 we have $\theta_r = \theta_{r'}$ and thus $r2 = r2^{'}$.
This establishes synchronization of events α_{s2} and α_{r2}.
Considering α_{r1}. Because$\langle\theta_r, \rho_r, \sigma_{r1}\rangle(ID_{ER})$ is secret, we can apply Lemma 3.27 of (21), which gives index $s1^{'} < r1$ such that
$\alpha_{s1'} = ((\theta_{s'}, \rho_{s'}, \sigma_{s1'}), send_1(s, r, \{|ID_S|\}k1(s,r), \{|TS_S|\}k1(s,r), m))$
$and \langle\theta_r, \rho_r, \sigma_{r1}\rangle(\{|ID_S|\}k1(S,R), \{|TS_S|\}k1(S,R), m)) = \langle\theta_{s'}, \rho_{s'}, \sigma_{s1'}\rangle(\{|ID_S|1(s,r), \{|TS_S|\}k1(s,r), m))$.
Correspondence of α_{s2} and α_{r2} gives,
$\langle\theta_s, \rho_s, \sigma_{s2}\rangle(TS_S) = \langle\theta_r, \rho_r, \sigma_{r2}\rangle(ID_{ER}) = \langle\theta_r, \rho_r, \sigma_{r1}\rangle(ID_{ER}) = \langle\theta_{s'}, \rho_{s'}, \sigma_{s1'}\rangle(TS_S)$.
By Lemma 3.28 θ_s and $\theta_{s'}$ are equal, which establishes synchronicity of events α_{r1} and α_{s1}.
This finishes the proof of Non-Injective synchronisation property of LAUP algorithm.

5 | SECURITY ANALYSIS BASED ON THREAT SCENARIOS:

The session key establishment and authentication method followed by LAUP algorithm are well suited for LoWPAN wireless network sensors. Because, the LAUP algorithm uses lightweight symmetric cryptographic methods to establish a session key and authentication process. Since the MAC address of the 6LoWPAN sensor device and Edge Router are in the encrypted form during the process of LAUP, it will not be disclosed to an eavesdropper. The proposed LAUP algorithm gives reliable protection against the well known LoWPAN security attacks.

REPLAY ATTACK: LAUP protects the transmission of messages from replay attack in all the four flights by adding MAC values, thereby the integrity of the message is maintained throughout the algorithm. So insertion, deletion or modification of messages could not be performed by the attacker. All the four flight information analyzed step by step for what would happen if the attacker captures the flight information. The first flight message packet could not be reproduced by the attacker because the packet contains information such as sensor$^{\prime}s$ unique MAC ID and the timer value of sensor at the time of first flight message generation and most importantly they are appended with MAC_{one} value. The second flight message contains the nonce value of Edge Router encrypted with the unique $flight_{two}$key SK2. Also, we proved the secrecy of Edge Router ID with Scyther tool, so that adversaries cannot get this information and reproduce it.

This strongly encrypted value cannot be deciphered by the attacker since he does not know the nonce value of Edge Router. The third flight message contains nonce values used in the first flight and the nonce value of the third flight, encrypted with unique $flight_{three}$key SK3. These ciphertexts are cryptographically strong enough for the lightweight communication and cannot be replayed so that the integrity of the message maintained. The fourth flight message has MAC_{four} value and ciphered form of the session key. An attacker can get the session key only if he knows unique $flight_{four}$ key SK4. On the whole, nonce values and MAC values prevent the attacker from replaying the message and maintaining integrity.

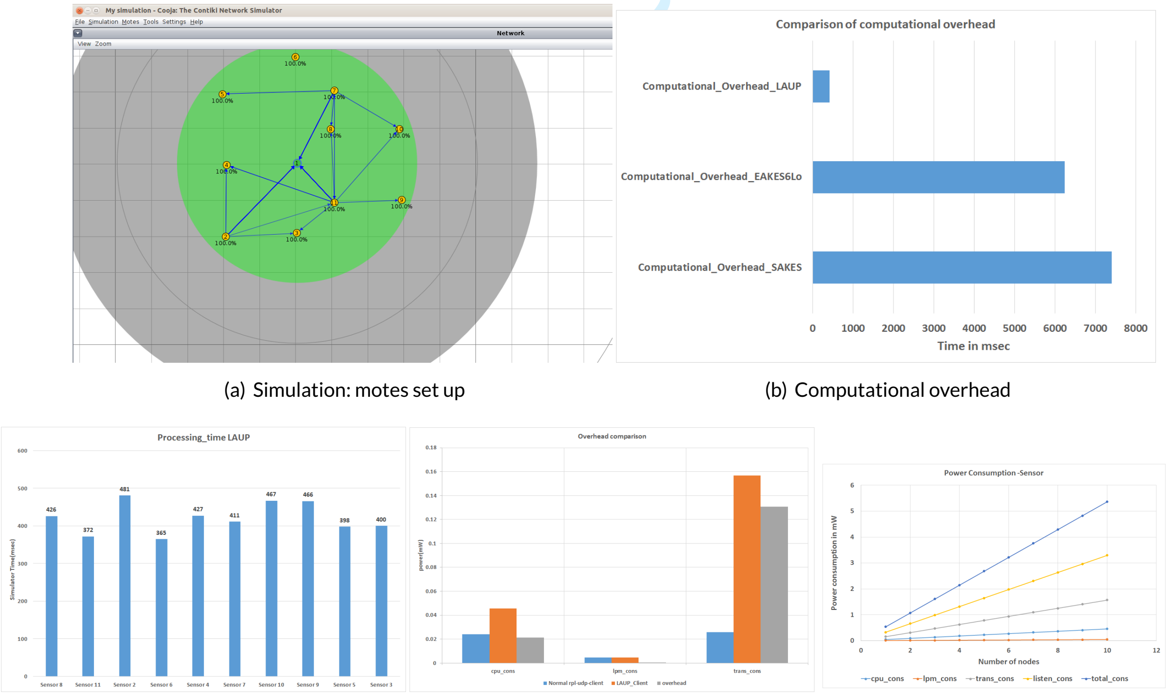
MAN IN THE MIDDLE ATTACK: LAUP protects the communication of messages against Man in the Middle attack. The man in the middle attacker possibly alters the communication between the two parties who believe that they are directly communicating with each other. But LAUP messages, in all four flights, are encrypted with the secure AES-128-ECB algorithm and unique flight keys.The flight messages are constructed with nonce values. Also, Non-Injective synchronization property is maintained.

IMPERSONATION ATTACK: Here an adversary can pretend like one of the legitimate sensors in the LoWPAN network. LAUP assumes all the sensors$^{\prime}$ identity are registered with the Edge Router. Sensor hello request from an impersonation adversary rejected by checking its identity.

6 | VALIDATION AND EVALUATION OF PROPOSED LAUP ALGORITHM

6.1 | Evaluation using COOJA simulator

Our proposed LAUP algorithm for authentication and key distribution algorithm simulated in Contiki OS COOJA simulator environment. Our simulated environmental architecture is shown in Figure 4(a). We have taken Wismote as a sensor and the Edge Router as well. The scalability of our proposed LAUP algorithm is checked by adding 65 nodes to the network with the Edge Router in the COOJA simulator. LAUP simulated like an RPL (Routing Protocol for Lossy networks) UDP client-server application whereas an Edge Router acts as a server.



(a) Simulation: motes set up

(b) Computational overhead

(c) Processing Time

(d) Power consumption

(e) Power consumption based on no.of nodes

FIGURE 4 The COOJA simulator - Evaluation results of LAUP.

Wismote voltage value and various current values such as CPU current value, low power mode current value, transmission, and current reception values are taken from the Wismote datasheet (24). Power consumption is calculated using the formula found in (25). The sensor who wants to communicate with the Edge Router is consuming 0.0456 mw of CPU power, 0.0048 mw of low power mode (lpm) power, 0.1567 mw of transmission power, 0.3300 mw of reception power and 0.5371 mw of total power. We simulated our algorithm with ten sensors. The graph in Figure 4(b) explains the comparison of computational overhead of LAUP with EAKES6Lo and SAKES (15) overhead values given in (9).

TABLE 3 Memory usage of the sensor and an Edge Router

text	data	bss	dec	hex	filename
45619	350	13236	59205	e745	udp-clientv1.wismote
49253	402	13782	63437	f7cd	border-routerv2.wismote

Although we compared the authentication algorithms (EAKES6Lo, SAKES) which simulated in different environments, the LAUP authentication algorithm provides 15 times less computational overhead than EAKES6lo and 18 times less computational overhead than SAKES authentication algorithm for LoWPAN devices. Figure 4(e) shows the power consumption value increases as the number of nodes increases also Figure 4(d) graph tells us power consumption while receiving messages is high compared to the lpm, and transmission energy consumption in LoWPAN devices.

The difference in power consumption of conventional sensor (without any authentication) and LAUP sensor (with the proposed authentication algorithm) is explained in Figure 4(d), and LAUP consumes 0.13079624 mw more power while transmitting messages than the regular sensor communication without authentication. Each flight of LAUP is communicated as a payload of the transport layer. Flight 1 (SensorHello) and Flight 2 (ERConf) consume 64 bytes each. Flight 3 (SensorChallenge) consumes 80 bytes. Flight 4 (Finished) consumes 48 bytes. Graph in Figure 4(e) reveals the total processing time of the LAUP algorithm for different sensors over time. From this graph, taking the average of the total processing time of 10 sensors, we proved that our proposed LAUP algorithm takes less time to execute the full authentication algorithm. Up to 65 6LoWPAN devices can be connected without resource-exhaustion to the Edge Router in a specific position. Coding of LAUP will be sent to the reader upon request.

Memory usage of LAUP algorithm is calculated on the sensor, and the Edge Router based on the information found in (26). Table 3 summarizes the memory usage of the sensor and the Edge Router. Data segment refers to read-write data, and bss segment indicates zero-initialized data. The sum of text, data and bss values mentioned in dec section. Flash consumption of LAUP algorithm is 45969 bytes in sensor and 49655 bytes in the Edge Router. RAM use of LAUP algorithm is 13586 bytes in sensor and 14184 bytes in the Edge Router. The total processing time of our proposed LAUP algorithm showed in Figure 4(c) takes 421.3 msec which is comparatively lower than the processing time of existing algorithms such as EAKES6Lo and SAKES are given in (9). This total processing time of LAUP is calculated form the COOJA simulators' mote output window.

6.2 | Hardware Evaluation

Figure 5 shows the testbed setup of the LAUP evaluation. We use our laptop with Contiki OS installed on VMware workstation 12 player, two wismotes from Arago systems, one MSP430 USB-Debug-Interface(MSP-FET430UIF) to upload the contiki program into wismote hardware and one TI CC2531 Dongle to capture 6LoWPAN packets. Wismote operates on 2.4GHz free band and supports IPV6. Sensors/actuators such as temperature, humidity, light or 3D accelerometer are available on WiSMote.

To upload our authentication program into wismote hardware, we have installed msp430flasher Linux version in Contiki OS. While uploading, MSPFlasher utility of Contiki OS invokes MSP430 USB-Debug-Interface(MSP-FET430UIF) to flash the Wismote memory. Once the Border router started, it prefixes aaaa to the link local address of the 6lowpan device which resides within the coverage of it and sets the IPv6 address. The 6LoWPAN device which needs to get authenticated by the Border Router has the same PAN ID as the Border router. The wismote sensor who wants to communicate with the Edge Router is consuming 0.0214 mw of CPU power, 0.0080 mw of low power mode (lpm) power and 0.0088 mw of transmission power of total power. A comparative power consumption of LAUP on wismote using the simulator and a hardware implementation is shown in Figure 6(a). Power consumption values such as CPU consumption, lpm consumption and transmission consumption are compared between simulation results and hardware results in Figure 6(a). LAUP authentication process running on wismote hardware takes 35 ticks or 1.0681 milliseconds to get connected with the border router. The same LAUP process running on simulated wismote takes 51 ticks or 1.5563 milliseconds which is comparatively less than the simulation result. Figure 6(b) and 6(c) clearly shows the difference between LAUP total processing time using simulated wismote using COOJA simulator and a testbed evaluation methods.

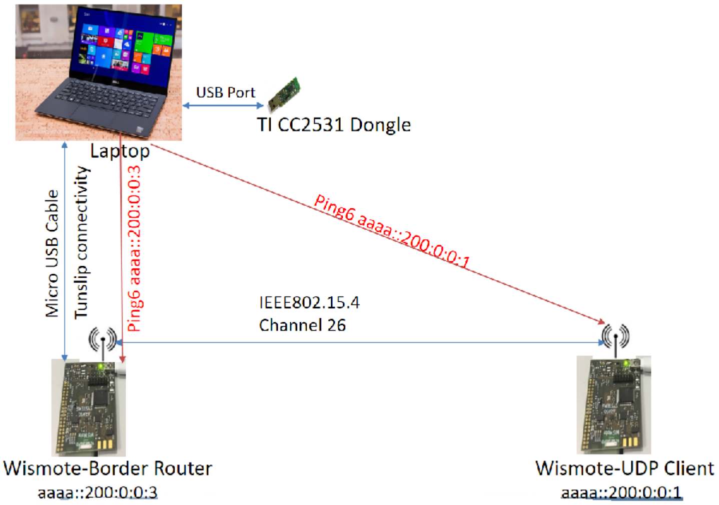
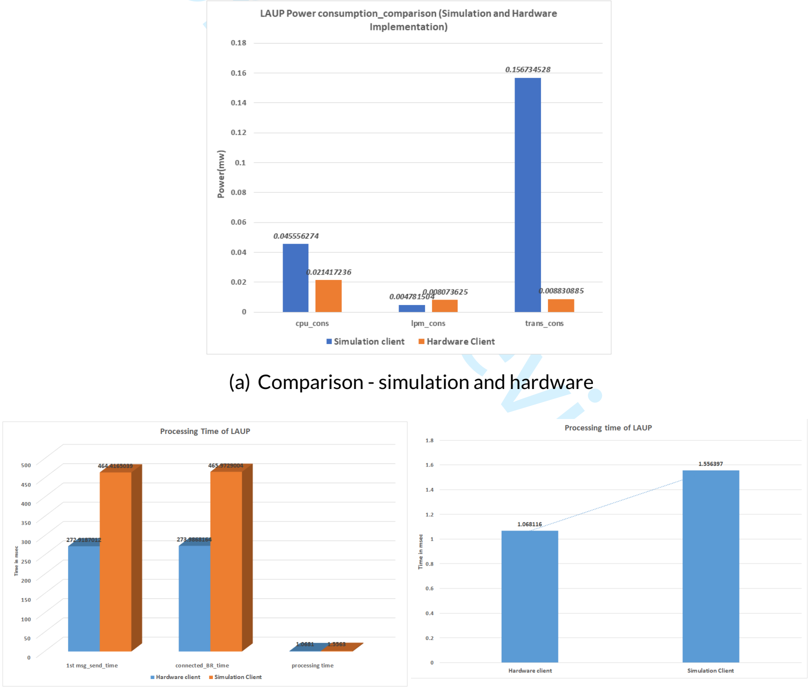


FIGURE 5 Hardware setup for LAUP



(a) Comparison - simulation and hardware

(b) Processing time of LAUP

(c) Processing time of LAUP

FIGURE 6 Test bed results on power consumption, and processing time of LAUP

7 | CONCLUSION

With the knowledge of existing algorithms and their limitations in the field of authentication and key distribution, we proposed our LAUP algorithm to overcome these limitations. Our algorithm formally verified by the formal verification tool called "Scyther", and we proved that authentication properties such as Aliveness, NonInjective Agreement, Secrecy of keys and Non-Injective Synchronization are maintained. Moreover, LAUP algorithm works with UDP protocol and possible threats such as Replay attack, Man in the Middle attack and impersonation attack are analyzed theoretically in section 5. In addition to the formal verification proof, we presented simulation results using the Contiki OS COOJA simulator with ten 6LoWPAN sensors as clients and one Edge Router. The hardware evaluation results broadly supported our simulation results and theoretical predictions. Evaluation of the proposed algorithm carried out based on the clock ticks of wismote. From our evaluation results, we can say that our algorithm is highly secured since LAUP generates the respective keys for each flight using the nonce value of sensors and Edge Router. Additionally, this LAUP algorithm is flexible to update the keys after each session. In future, LAUP will be tested against various attacks such as Sybil attacks, DoS attacks and replay attacks using Cooja simulator and the hardware. From the verification tool results, evaluation results from the simulator and hardware, we proved that the LAUP algorithm for authentication and key distribution is faster, highly secured, scalable for LoWPAN networks and flexible enough to update the keys dynamically.

ACKNOWLEDGMENTS

We would like to thank NICTA and DAT61 for their support and funding for this project, Project no: 1032350.

References

[1] Shelby Zach, Bormann Carsten. *6LoWPAN: The wireless embedded Internet*. John Wiley & Sons; 2011.

[2] Mahmoud Rwan, Yousuf Tasneem, Aloul Fadi, Zualkernan Imran. Internet of things (IoT) security: Current status, challenges and prospective measures. In: :336-341IEEE; 2015.

[3] Al-Fuqaha Ala, Guizani Mohsen, Mohammadi Mehdi, Aledhari Mohammed, Ayyash Moussa. Internet of things: A survey on enabling technologies, protocols, and applications. *Communications Surveys & Tutorials, IEEE*. 2015;17(4):2347-2376.

[4] Sheng Zhengguo, Yang Shusen, Yu Yifan, Vasilakos Athanasios, Mccann Julie, Leung Kin. A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wireless Communications*. 2013;20(6):91–98.

[5] Gheorghe Laura, Rughinis Razvan, Deaconescu Razvan, Tapus Nicolae. Authentication and anti-replay security protocol for wireless sensor networks. In: :7–13IEEE; 2010.

[6] Jara Antonio J, Marin Leandro, Skarmeta Antonio FG, Singh Dhananjay, Bakul Gohel, Kim Daeyeoul. Mobility modeling and security validation of a mobility management scheme based on ECC for IP-Based Wireless Sensor Networks (6LoWPAN). In: :491-496IEEE; 2011.

[7] Al Fardan Nadhem J, Paterson Kenneth G. Lucky thirteen: Breaking the TLS and DTLS record protocols. In: :526-540IEEE; 2013.

[8] Raza Shahid, Voigt Thiemo, Jutvik Vilhelm. Lightweight IKEv2: a key management solution for both the compressed IPsec and the IEEE 802.15. 4 security. In: Citeseer; 2012.

[9] Qiu Yue, Ma Maode. A Mutual Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN Networks. *IEEE Transactions on Industrial Informatics*. 2016;12(6):2074–2085.

[10] Esfahani Alireza, Mantas Georgios, Matischek Rainer, et al. A lightweight authentication mechanism for M2M communications in industrial IoT environment. *IEEE Internet of Things Journal*. 2017;.

[11] Raza Shahid, Trabalza Daniele, Voigt Thiemo. 6LoWPAN compressed DTLS for CoAP. In: :287–289IEEE; 2012.

[12] Krentz Konrad-Felix, Rafiee Hosnieh, Meinel Christoph. 6LoWPAN security: adding compromise resilience to the 802.15. 4 security sublayer. In: :1ACM; 2013.

[13] Perrig Adrian, Szewczyk Robert, Tygar Justin Douglas, Wen Victor, Culler David E. SPINS: Security protocols for sensor networks. *Wireless networks*. 2002;8(5):521–534.

[14] Krentz Konrad-Felix, Meinel Christoph. Handling reboots and mobility in 802.15. 4 security. In: :121–130ACM; 2015.

[15] Hussen Hassen Redwan, Tizazu Gebere Akele, Ting Miao, Lee Taekkyeun, Choi Youngjun, Kim Ki-Hyung. SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6L0WPAN). In: :246–251IEEE; 2013.

[16] Li Ming, Yu Shucheng, Guttman Joshua D, Lou Wenjing, Ren Kui. Secure ad hoc trust initialization and key management in wireless body area networks. *ACM Transactions on sensor Networks (TOSN)*. 2013;9(2):18.

[17] Pietro Roberto Di, Ma Di, Soriente Claudio, Tsudik Gene. Self-healing in unattended wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*. 2012;9(1):7.

[18] Premnath Sriram N, Haas Zygmunt J. Security and Privacy in the Internet-of-Things Under Time-and-Budget-Limited Adversary Model. *Wireless Communications Letters, IEEE*. 2015;4(3):277-280.

[19] Hennebert Christine, Dos Santos Jessye. Security protocols and privacy issues into 6lowpan stack: A synthesis. *Internet of Things Journal, IEEE.* 2014;1(5):384-398.

[20] Mukherjee Amitav. Physical-Layer Security in the Internet of Things: Sensing and Communication Confidentiality Under Resource Constraints. *Proceedings of the IEEE.* 2015;103(10):1747-1761.

[21] Cremers Casimier Joseph Franciscus. *Scyther: Semantics and verification of security protocols.* Eindhoven University of Technology; 2006.

[22] Lowe Gavin. A hierarchy of authentication specifications. In: :31–43IEEE; 1997.

[23] Cremers Cas, Mauw Sjouke. Operational semantics of security protocols. In: Springer 2005 (pp. 66–89).

[24] Instruments Texas. *CC2520 DATASHEET, 2007.* 2014.

[25] SonHan . Thigschat Internet of things http://thingschat.blogspot.com.au/2015/04/contiki-os-using-powertrace-and.html/[Online; accessed 12-November-2016]; 2016.

[26] Velinov Aleksandar, Mileva Aleksandra. Running and Testing Applications for Contiki OS Using Cooja Simulator. 2016;.

AUTHOR BIOGRAPHY

Annie Gilda Roselin received **ME** degree in Computer Science and Engineering from Mepco Schlenk Engineering College, Sivakasi, Anna University, Chennai and **B.Tech** degree in Information Technology from Noorul Islam College of Engineering, Anna University, Chennai. She has worked as a Lecturer in Computer Science and Engineering department of Velammal Engineering College, Chennai, India for three years from 2008 to 2011. She is currently doing Ph.D. degree in 6LoWPAN protocol security from School of Electrical and Data Engineering, University of Technology, Sydney.

Dr.Priyadarsi Nanda is a Senior Lecturer at the University of Technology Sydney (UTS) with extensive experience in research and development of Cyber Security, IoT security, and wireless sensor network security. His most significant work has been in the area of Intrusion Detection and Prevention systems using image processing techniques, Sybil attack detection in IoT based applications, intelligent firewall design. He has published 80+ high quality refereed research papers including Transactions on Computers, Transactions in Parallel Processing and Distributed Systems, Future Generations of Computer Systems (FGCS). He has successfully supervised 10 research students in the past and currently supervising 7 research students in Cybersecurity research. Dr. Nanda holds a PhD in Computing Science, Master in Computer Engineering and Bachelor in Computer Engineering.

Dr.Surya Nepal is a Principal Research Scientist at Information Engineering Laboratory of CSIRO Computational Informatics. Dr. Nepal has 15+ years experience in computer science research, latterly with a specific focus on security, privacy and trust in distributed systems. He has more than 100 publications to his credit, has edited or co-authored several books, and is the co-inventor of two patents. Much of his work appears in top international forums such as VLDB, ICDE, ICWS, SCC, CoopIS, ICSOC, International Journals of Web Services Research, IEEE Transactions on Service Computing, IEEE TPDS, ACM Computing Survey and ACM Transaction on Internet Technology. He obtained his BE from the National Institute of Technology, Surat, India; his ME from the Asian Institute of Technology, Bangkok, Thailand; and Ph.D. from RMIT University, Australia.

Dr.Sean He as a Chief Investigator, has received various research grants including four national Research Grants awarded by Australian Research Council (ARC). He is the Director of Computer Vision and Pattern Recognition Laboratory at the Global Big Data Technologies Centre (GBDTC) at the University of Technology Sydney (UTS). He is an IEEE Senior Member and has been an IEEE Signal Processing Society Student Committee member. He has been awarded 'Internationally Registered Technology Specialist' by International Technology Institute (ITI). He is a leading researcher in several research areas including big-learning based human behaviour recognition on a single image, image processing based on hexagonal structure, authorship identification of a document and a document's components (e.g., sentences, sections etc.), network intrusion detection using computer vision techniques, car license plate recognition of high speed moving vehicles with changeable and complex background, and video tracking with motion blur.

□