Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/JIFS-219422
IOS Press

One-key based cryptographically generated address for location update in next generation IP mobility

Senthilkumar Mathi[a,*], Uma Jothi[a], G. Saravanan[b], Venkadeshan Ramalingam[c] and K. Sreejith[a]
[a]*Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore, Amrita Vishwa Vidyapeetham, India*
[b]*Department of Artificial Intelligence and Data Science, Erode Sengunthar Engineering College, Erode, India*
[c]*Faculty in Information Technology Department, University of Technology and Applied Sciences-Shinas, Sultanate of Oman*

Abstract. Mobile devices have risen due to internet growth in recent years. The next generation of internet protocol is evolving for mobile devices to generate their addresses and get continuous services across networks to support the enormous number of addresses in network-based mobility. The mobile device updates its current location to its home network and the correspondent users through a binding update scheme in the visited network. Numerous studies have investigated binding update schemes to verify the reachability of the mobile device at its home network. However, most schemes endure security threats due to the incompetence of authenticating user identity and concealing the temporary location of mobile devices. To address these issues, this paper proposes a secure and efficient binding update scheme (One-CLU) by incorporating a one-key-based cryptographically generated address (CGA) to validate and conceal the address ownership of mobile devices with minimal computations. The security correctness of the proposed One-CLU scheme is verified using AVISPA – a model checker. Finally, the simulation and the numerical results show that the proposed scheme significantly reduces communication payloads and costs for the binding update, binding refresh, and packet delivery.

Keywords: Mobile communication, routing, privacy, cryptography, communication security

1. Introduction

The Mobile Internet Protocol (MIP) provides seamless communication between nodes by attaining its respective internet protocol (IP) address. When the node enters a new location, its IP address changes accordingly and acquires its new address for further communication [1]. Hence, the mobility-based protocol for IPv6 called MIPv6 is envisioned to allow the host to enter different networks without losing connection. The main entities of MIPv6 are the mobile node (MN), the home agent (HA), and the correspondent node (CN). The MN is the mobile host that uses its permanent home-of address (HoA) to communicate with its home network.

When the MN enters its new network, it acquires the temporary address called care-of address (CoA) to maintain communication with its respective CN through the HA, as shown in Fig. 1. Thus, the MN establishes the bidirectional tunnel [2] with its HA after moving to the other network. Hence, the HA intercepts the message destined for the MN's HoA and is forwarded to the MN's current location (CoA). Similarly, reverse tunneling sends the

[*]Corresponding author. Senthilkumar Mathi, Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore, Amrita Vishwa Vidyapeetham, India. E-mail: m_senthil@cb.amrita.edu.

ISSN 1064-1246/$35.00 © 2024 – IOS Press. All rights reserved.

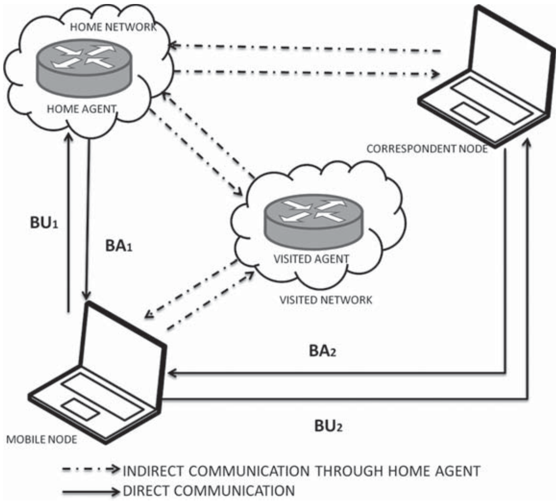


Fig. 1. Direct and indirect communication in MIPv6.

message from MN to CN. It increases the problem of inefficient routing since bidirectional tunneling adopts the encapsulation method to perform the tunneling between MN and the HA [3]. To overcome the drawbacks of bidirectional tunneling, triangular routing was suggested that sends the packets directly from MN to the CN but not vice versa. It does not provide an optimized route since the message from CN to the MN flows through HA, thus resulting in a longer path [4].

Further, the direct communication between MN and CN is suggested with Ren et al.'s route optimization (RO) method. This method requires the binding cache at the CN for storing the MN's HoA and CoA; hence, the message from the CN is forwarded to the MN's CoA instead of HoA [5]. Thus, the MN and CN exchange the packets directly by sending the binding update (BU) messages and binding acknowledgment (BA). However, security threats exist because the attacker captures the BU messages to launch attacks such as man-in-the-middle (MITM), replay, false BU, and denial-of-service [6, 7].

Most schemes suffer from security vulnerabilities and latency issues due to the incompetence of the balancing effort between security and efficiency [8–12]. They are suffering from security threats, increased computation costs, complexity, and inability to validate address ownership. This motivates us to propose a new BU scheme using one-key-based CGA to enhance the efficacy and security of BU in MIPv6.

Hence, the current paper suggests a new binding update scheme using one-key-based CGA to augment the efficacy and security of BU in MIPv6. The contributions of the paper are as follows: 1) generating and verifying CoA, 2) providing mutual authentication between participating principals, 3) concealing the MN's identity, 4) validating the address ownership of the MN at CN to prevent rogue nodes, 5) verifying the correctness of the security properties of the proposed scheme using Automated Validation of Internet Security Protocols and Application (AVISPA) 6) significant reduction in the number of signaling messages and computational complexity to enhance efficiency.

2. Related work

Ali Alsalihy et al. have emphasized return routability using identity-based encryption (RRIBE) that involves a third party generating the private keys for the mobile hosts [8]. The RRIBE requires a third party, a private key generator (PKG), to generate and distribute the keys to its respective mobile devices [8]. Initially, the MN sends the packet to CN through HA and directly to CN. On receiving both packets, the CN verifies whether the received contents of the packets are similar. If both are the same, the PKG generates the private key to provide authentication between the participating principals.

Similarly, other packets are generated and used to authenticate the MN and CN nodes. But, if the PKG is compromised, an illegitimate user can easily acquire all the keys and information stored in its database. In addition, if the PKG loses all the secret keys, key revocation becomes impossible. The batch binding update (BBU) scheme for RO aims at verifying the batch of binding updates simultaneously at the receiver [9]. An additional mobile router is deployed, connecting all the MNs within the network, ensuring that the mobile router carries out the route optimization instead of MN. It uses a multi-key cryptographically generated address for generating the MN's CoA to ensure address ownership on the verifier side. Since it is mobile, it frequently moves into a new network, verifying that binding updates in one network becomes impossible. However, the generation of private keys using a third party is not secure as it rises to vulnerabilities. In addition, it suffers from a time-memory trade-off attack since the possibility of overflow at the stack of CN is high.

Successively, to improve the efficiency and to prove the address ownership at the respective CN, a method known as secure route optimization for MIPv6 using enhanced cryptographically generated

Table 1
Comparative analysis of state-of-art

Feature	RRIBE	BBU	CGA and DNSSEC	Private Key-Based BU	TOTP
Third-party dependency	Yes	No	Yes	No	No
Security	Vulnerable to PKG compromise	Secure	Secure	Vulnerable to private key disclosure	Secure
Efficiency	High computational cost	High computational cost	High computational cost	Moderate computational cost	Moderate computational cost
Address ownership verification	Yes	Yes	Yes	No	No
Message overhead	High	High	High	Moderate	Moderate
Complexity	Yes	Yes	Yes	Yes	Yes
Key management	PKG-based	Multi-key CGA	CGA and DNSSEC	MN's private key	Shared secret token
Suitability	Suitable for networks with low mobility	Suitable for networks with high mobility	Suitable for networks with low mobility and trusted DNS servers	Suitable for networks with low mobility and trusted private key management	Suitable for networks with high mobility and low computational resources

address (CGA) and domain name system security extension (DNSSEC) was proposed [10]. Here, the MN in the visited network receives the home token from HA for future communication and adopts the CGA method to generate its new address (CoA). The HA and visited HA validate the token for authentication to check whether the RO is allowed for the corresponding MN. As a result, it requires a DNS server each time by the HA and visited HA to verify the domain name. In addition, the CN employs DNSSEC to ensure that the MN's HoA and HA are in the trusted domain. Here, both participants require DNSSEC, which increases the cost of implementation, and thus, the overall efficiency decreases. In addition, it requires more computation time to verify the CoA generated using the CGA method as it involves a backward hash chain.

Later, the private key-based binding update scheme was examined to validate the address ownership of the MN [11]. The MN in its visited network generates CoA by applying the hash function on its private key. It provides the CN to ensure that the MN is a valid node and reduces the number of BUs. Nevertheless, security issues arise since the MN uses its private key to generate CoA. Thus, if the MN's private key is shared, all the information passed to the MN is easily captured. However, it suffers from security threats of using its private key as its CoA.

Consecutively, a time-based one-time password (TOTP) method was examined to reduce the handover delay [12]. It generates the shared secret token using the opponent's randomly generated number and public key. It sends the node's status request, including its public key information, from MN to check the reachability of the CN. However, it does not provide validation of the address ownership of the MN. The CN cannot validate the address ownership, though it reduces the signaling overhead.

Table 1 highlights the comparative analysis of existing techniques like reliance on a third party, security vulnerabilities, increased computation costs, and inability to validate address ownership. Therefore, considering the problems of all the earlier works, the current paper proposes a secure and optimized scheme (One-CLU) where the MN incorporates a one-key-based CGA method to provide mutual authentication for the nodes.

3. Proposed one-CLU scheme

The intricacies of the proposed One-CLU scheme are discussed in this section. The notations used in the proposed scheme are listed in Table 2. The main aim of the proposed scheme is to validate the address ownership of the mobile user in terms of secured and efficient signaling messages. Here, the mutual authentication of the communicants is carried out by generating the CoA of MN using a one-key-based CGA technique. The following sub-sections discuss the initial registration of MN at the home network, the key generation of MN, HA, and CN, and the detailed description of the proposed BU scheme.

Table 2
Notations used in the proposed One-CLU scheme

Notation	Description
$H(F_1 \parallel F_2)$	Hash value with combined fields F_1 and F_2
$\{F_1, F_2\}K$	Encryption of fields F_1 and F_2 using key K
MN_{HoA}	Home-of address of MN
HA_{Addr}	Address of HA
MN_{Rand}, HA_{Rand}, CN_{Rand}	Random number of MN, HA, and CN
MN_{Pub}, HA_{Pub}, CN_{Pub}	Public key of MN, HA, and CN
MN_{Pri}, HA_{Pri}, CN_{Pri}	Private key of MN, HA, and CN
$SK_{HA\text{-}MN}$	Secret key shared between HA and MN
$SK_{MN\text{-}CN}$	Secret key shared between MN and CN
MN_{N0}, HA_{N0}, CN_{N0}	Nonce of MN, HA, and CN
MN_{N1}, HA_{N1}	New nonce of MN and HA
MN_{Sig}	Signature of MN

3.1. Initial association of MN with home network

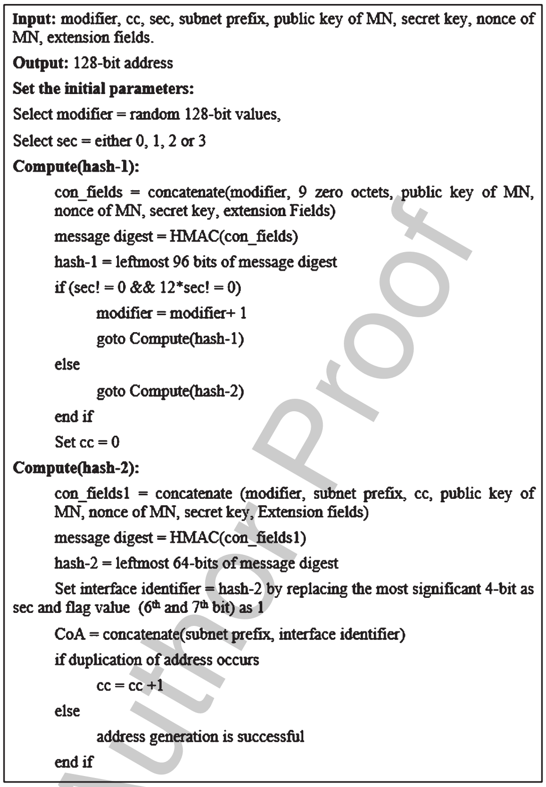
Initially, HA shares the parameters such as MN_{HoA}, HA_{Addr}, G_P, MN_{N0}, $HA_{N0,}$ $CoA_{param,}$ and HA_{Pub} with the MN. The MN stores these values in its dynamic parameter database for further communication. In addition, HA shares the modifier value with the MN for generating CoA when the MN enters a new network.

3.2. A detailed description of the binding update by MN with HA after relocation

The step-by-step intricacies of the proposed BU with HA are discussed as follows,

Step 1: CoA Generation by MN in its visited network

Initially, the MN concatenates the modifier value with the 72-bit zeros, the public key of MN, nonce of MN, secret key, security flag *sec*, where $0 < sec < 8$, and any required extension fields. The most significant 96 bits are generated as the output of *hash-2* by applying the hash-based message authentication code (HMAC) function with the concatenated values, as shown in Fig. 2. The computation of *hash-2* is repeated until its leftmost $12 \times sec$ bits obtained the value '*0*' by incrementing the last bit of modifier value. The *hash-1* value is computed by applying HMAC on the concatenation of modifier, subnet prefix, collision count (*cc*), the public key of MN, nonce of MN, and the secret key. Subsequently, the leftmost 64 bits from the *hash-1* are retrieved as the *interface identifier* by replacing the most significant 4 bits with the *sec* value and fixing the flag value to *1*. Finally, the CoA is formed by concatenating the subnet prefix and the generated interface identifier. After generating CoA, the MN sends the binding update message to HA.



```
Input: modifier, cc, sec, subnet prefix, public key of MN, secret key, nonce of MN, extension fields.
Output: 128-bit address
Set the initial parameters:
Select modifier = random 128-bit values,
Select sec = either 0, 1, 2 or 3
Compute(hash-1):
    con_fields = concatenate(modifier, 9 zero octets, public key of MN, nonce of MN, secret key, extension Fields)
    message digest = HMAC(con_fields)
    hash-1 = leftmost 96 bits of message digest
    if (sec! = 0 && 12*sec! = 0)
        modifier = modifier+ 1
        goto Compute(hash-1)
    else
        goto Compute(hash-2)
    end if
    Set cc = 0
Compute(hash-2):
    con_fields1 = concatenate (modifier, subnet prefix, cc, public key of MN, nonce of MN, secret key, Extension fields)
    message digest = HMAC(con_fields1)
    hash-2 = leftmost 64-bits of message digest
    Set interface identifier = hash-2 by replacing the most significant 4-bit as sec and flag value (6th and 7th bit) as 1
    CoA = concatenate(subnet prefix, interface identifier)
    if duplication of address occurs
        cc = cc +1
    else
        address generation is successful
    end if
```

Fig. 2. Pseudocode for CoA generation.

Step 2: Location update of MN

(M1) MN → HA:

HA_{Addr}, MN_{CoA}, $MN_{N0,}$ $HMAC(MN_{CoA} \| HA_{N0})$, $\{MN_{CoA}, MN_{HoA}, MN_{N0}\}$ HA_{Pub}

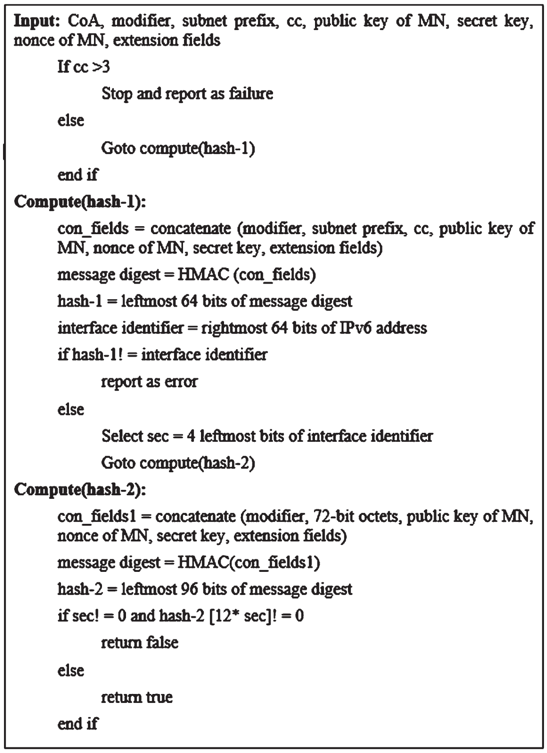
After generating the new $CoA(MN_{CoA})$, the MN computes the HMAC function with the concatenation of MN_{CoA} and HA_{N0}. In addition, it encrypts the parameters such as MN_{CoA}, $MN_{HoA,}$ and MN_{N0} using HA_{Pub} and sends all the fields as the BU message to the HA.

Step 3: Upon receipt of location update (M1) at HA

- Verifies the HMAC value by generating its secret key ($SK_{HA\text{-}MN}$).
- Decrypts the message using its private key (HA_{Pri}).

Verifies the CoA by using the one-key-based CGA.

Step 4: Verification of CoA by HA



```
Input: CoA, modifier, subnet prefix, cc, public key of MN, secret key,
nonce of MN, extension fields
    If cc >3
        Stop and report as failure
    else
        Goto compute(hash-1)
    end if
Compute(hash-1):
    con_fields = concatenate (modifier, subnet prefix, cc, public key of
    MN, nonce of MN, secret key, extension fields)
    message digest = HMAC (con_fields)
    hash-1 = leftmost 64 bits of message digest
    interface identifier = rightmost 64 bits of IPv6 address
    if hash-1! = interface identifier
        report as error
    else
        Select sec = 4 leftmost bits of interface identifier
        Goto compute(hash-2)
Compute(hash-2):
    con_fields1 = concatenate (modifier, 72-bit octets, public key of MN,
    nonce of MN, secret key, extension fields)
    message digest = HMAC(con_fields1)
    hash-2 = leftmost 96 bits of message digest
    if sec! = 0 and hash-2 [12* sec]! = 0
        return false
    else
        return true
    end if
```

Fig. 3. Pseudocode for CoA verification.

The HA verifies the received CoA and ensures that it validates the address ownership of MN. The pseudocode for verifying the CoA is shown in Fig. 3. Initially, the HA/CN retrieves the *cc* value (i.e., 0, 1, or 2) and checks the subnet prefix from the CGA parameter equivalent to the sender's address. If it is not equal, report it as an error; otherwise, it takes the leftmost 64 bits of the address, known as the interface identifier. Subsequently, the HA computes the *hash-1* value by applying the HMAC function on the concatenation of modifier, subnet prefix, cc, public key of MN, nonce of MN, and the secret key. The most significant 64 bits from the *hash-1* value are compared with the *interface identifier* of the address by leaving the *sec* value and flag value. If equivalent, the HA computes the *hash-2* value by applying the HMAC function on the concatenated values such as the modifier, nine octets, public key of MN, nonce of MN, and the secret key generated using the ECDH scheme. Consequently, retrieve the *sec* value from the *interface identifier* of the sender's address and compare it to the *12* × *sec* bits of the most significant bits of *hash-2*. If *12* × *sec* values are zero, the verification process is successful; otherwise, stop and report as failed.

Step 5: HA sends the binding acknowledgment to MN

(M2) HA → MN: MN_{CoA}, HA_{Addr}, HMAC(MN_{N1} ||HA_{N0}), {MN_{N1}, HA_{N1}}MN_{Pub}.

Subsequently, the HA generates the new nonce value for MN (MN_{N1}) and HA (HA_{N1}) and transmits it to MN by encrypting with MN_{Pub}. It also attaches the hash value (applying a hash function on the nonce MN_{N1} and HA_{N1}) to provide authentication and data integrity.

Step 6: Upon receipt of M2 at MN

- Computes the HMAC using $SK_{HA\text{-}MN}$ and checks whether the received HMAC value and the computed HMAC value are equal.
- The new nonce value, such as MN_{N1} and HA_{N1}, is retrieved by decrypting M2 with the MN_{Pri}.

3.3. Binding update by MN with CN

Step 1: Sending binding update message from MN to CN

(M3) MN → CN: CN_{Addr}, MN_{CoA}, {MN_{CoA}, $MN_{N1,}$ $MN_{Pub,}$ CoA_{param}}CN_{pub}, MN_{Sig}(HMAC(MN_{CoA}|| MN_{N1}))After registering MN_{CoA} with the HA, the MN sends a BU message to CN by attaching all the fields required for the CN. It encrypts the packet containing MN_{CoA}, $MN_{N1,}$ $MN_{Pub,}$ and CoA_{param}. It also adds its signature (MN_{sig}) to the HMAC function containing MN_{CoA} and MN_{N1}.

Step 2: Upon receipt of M3 at CN

- Decrypts the message sent by MN using CN_{Pri} to retrieve the CoA_{param} and other values.
- Verifies the CoA using a one-key-based CGA method.
- Validates the MN_{sig} using MN_{Pub}.

Step 3: BA message from CN to MN

(M4) CN → MN: MN_{CoA}, $CN_{Addr,}$ HMAC(CN_{N0}||MN_{N1}), {CN_{N0}, MN_{N1}}$MN_{Pub.}$The CN verifies the CoA and sends the BA message that contains the HMAC value. Besides, it appends the fields CN_{N0} and MN_{N1} by encrypting with the public key of MN (MN_{Pub}).

Step 4: Upon receipt of M4 at MN

- Decrypts and retrieves the value of CN_{N0} using MN_{Pri}.
- Computes the HMAC function by concatenating CN_{N0} and MN_{N1} and validates it with the received HMAC.

4. Security analysis and formal verification using AVISPA – A model checker

This section discusses the security features of the proposed BU scheme with the existing schemes. In addition, the validation of the proposed One-CLU scheme using AVISPA is discussed.

4.1. Authentication

Authentication involves the process of verifying the user’s identity in communication. It ensures that all BUs received by the entities are from the valid user. It can be provided by cryptographic authentication functions such as message authentication code, hash functions, and HMAC. The comparative analysis of authentication with existing schemes and the proposed BU scheme are listed in Table 3. In the proposed One-CLU scheme, the receiver verifies the MN’s CoA to ensure that the MN is a valid node [14]. Consequently, the CoA verification includes the HMAC with a secret key known to the legal communicants, and the attacker fails to recover the secret key.

4.2. Confidentiality

Confidentiality aims at protecting the secrecy of data from unauthorized access. The proposed scheme provides confidentiality by encrypting the messages with the public key of the participants, such as {MN_{CoA}, $MN_{N1,}$ $MN_{Pub,}$ CoA_{param}}CN_{pub} and {CN_{N0}, MN_{N1}} $MN_{Pub.}$ Hence, the valid user (CN/MN) only decrypts the message using the private key to view the message. Table 4 shows the confidentiality analysis.

4.3. Data integrity

It checks whether the intruder modifies the data flow between the participants during communication. The proposed One-CLU scheme uses the HMAC function for integrity checks. The sender (CN/HA) attaches the message HMAC (CN_{N0}||MN_{N1}), and HMAC (MN_{N1} || HA_{N0}) to the MN, and thus the receiver (MN) verifies the message by computing the HMAC function. If the computed hash message is equivalent to the received hash value, then the recipient accepts the message or rejects the data. The analysis of integrity is shown in Table 6.

Table 3
Authentication analysis

Scheme	MN-HA	HA-MN	MN-CN	CN-MN
RRIBE	Tunnelling	Tunnelling	Ŋ	Ŋ
BBU	Tunnelling	Tunnelling	$	$
RO-DNSSEC	$	Ø	$	$
PKBU	$	Ŋ	$	$
TOTP	$	$	$	$
One-CLU	$	$	$	$

$ = provided; Ŋ = Not provided; Ø = Not considered.

Table 4
Confidentiality analysis

Scheme	MN-HA	HA-MN	MN-CN	CN-MN
RRIBE	Tunnelling	Tunnelling	$	$
BBU	Tunnelling	Tunnelling	$	$
RO-DNSSEC	$	Ø	$	$
PKBU	$	Ŋ	$	$
TOTP	Ŋ	Ŋ	$	$
One-CLU	$	$	$	$

$ = provided; Ŋ = Not provided; Ø = Not considered.

Table 5
Non-repudiation analysis

Scheme	MN-HA	HA-MN	MN-CN	CN-MN
RRIBE	Tunnelling	Tunnelling	Ŋ	Ŋ
BBU	Tunnelling	Tunnelling	$	$
RO-DNSSEC	$	Ø	$	$
PKBU	Ŋ	Ŋ	$	$
TOTP	$	$	$	$
One-CLU	$	$	$	$

$ = provided; Ŋ = Not provided; Ø = Not considered.

Table 6
Attack prevention analysis

Scheme	MN-HA	HA-MN	MN-CN	CN-MN
RRIBE	Tunnelling	Tunnelling	$	$
BBU	Tunnelling	Tunnelling	$	$
RO-DNSSEC	$	Ø	$	$
PKBU	$	Ŋ	$	$
TOTP	Ŋ	Ŋ	$	$
One-CLU	$	$	$	$

$ = provided; Ŋ = Not provided; Ø = Not considered

4.4. Replay attack prevention

A replay attack is a malicious threat to the network. A valid transmitted message from the source address is repeated fraudulently or delayed for a certain time [15]. The proposed One-CLU scheme would be avoided by appending the randomly generated nonce with both MN and CN messages. Hence, if the intruder tries to intercept the message, the nonce changes immediately; thus, the verifier fails to validate the message and drops it.

4.5. Denial-of-service attack

The attacker explicitly accomplishes the Denial-of-Service (DoS) attack to make the resource unavailable or create network traffic. It is carried out by continuously sending false binding update messages to the target node. The proposed One-CLU scheme prevents this attack by verifying the CoA using the One-key-based CGA method to check the ownership of the address. Thus, if the address verification fails, the HA or CN ignores the packet.

4.6. Amplification attack prevention

Amplification is a distributed Dos attack in which the attacker repeatedly delivers the network traffic to the victim node and temporarily interrupts its resources. Here, the digital signature can be applied to the message while transmitting to eliminate amplification attacks. In the proposed One-CLU scheme, the MN attaches its signature with the packet (i.e., $MN_{Sig}(HMAC(MN_{CoA}||MN_{N1}))$) to the CN/HA for authenticating the message. Hence, if the signature is valid, the CN accepts the packet and proceeds with the next level; otherwise, it discards the message.

4.7. False binding update prevention

The false binding update intercepts the BU messages and replaces them with the new message. Therefore, the messages in the data packet can easily be changed by the intruder and forwarded to valid users. However, our proposed One-CLU scheme eliminates the false binding update attack by adopting a secret key in CoA generation since it involves the user's private key. The attacker cannot generate the secret key. Hence, the attacker fails to prove it is a valid node to the communicant.

4.8. MITM attack prevention

The attacker listens to the communication between two nodes and changes the content of the data packet [16]. It is not possible in the proposed scheme since the attacker does not know the user's secret and private keys to modify the messages' content. Accordingly, if the intruder tries to send a fake BU from its address to the CN or HA, it fails to provide authenticity, integrity, and address ownership of the message.

Table 7
Attack prevention analysis

Attack	RRIBE	BBU	PKBU	TOTP	One-CLU
False BU	Ŋ	$	Ŋ	Ŋ	$
MITM	$	$	$	$	$
DoS	Ŋ	$	$	$	$
Session-Hijack	Ŋ	$	$	Ŋ	$
Replay	$	$	Ŋ	$	$
Amplification	$	Ŋ	Ŋ	$	$

$ = provided; Ŋ = Not provided; Ø = Not considered.

4.9. Session hijacking attack prevention

In session hijacking, the intruder claims its address or fake address as the MN's CoA and forwards it to the CN or HA. If the receiver does not check for the message's authenticity, the CN or HA sends the valid information to the attacker instead of the MN. However, our One-CLU scheme thwarts the hijacking attack by verifying the CoA at the receiver. Only the valid user knows the CoA parameters and the secret key used in the one-key-based CGA generation and HMAC. The receiver accepts the BU or BA message or drops it if the verification is successful. Table 7 compares security features on various binding update schemes for their strengths and weaknesses.

4.10. Formal verification using AVISPA – A model checker

The AVISPA tool is primarily used to verify the security features of protocols. The protocol uses High-Level Protocol Specification Language (HLPSL) [17]. It is then converted into a low-level format called Intermediate Format (IF) using the translator known as hlpslif. Successively, the IF forwarded to the back-ends, such as On-The-Fly-Model-Checker (OFMC) and Constraint-logic-based attack searcher (CL-AtSe), to trace the sequence of events [18]. The simulation results of the One-CLU scheme are verified using back-end OFMC and CL-AtSe for the security properties, as shown in Fig. 4. As a result, no revealed attacks are shown from the execution trace of the proposed scheme.

The security analysis utilizes rigorous formal verification via the AVISPA model checker, providing cryptographic protocol guarantees stronger than empirical simulations. The analysis considers a comprehensive set of relevant attack vectors across authentication, confidentiality, integrity, availability, and system security properties. Each claimed security feature is logically argued based on the specific cryptographic mechanisms and protocol operations employed. Standards-compliant cryptographic primitives formally establish properties like authentication, integrity, non-repudiation, and resistance to spoofing. Security reductions demonstrate breaking the scheme requires breaking the underlying cryptographic assumptions. The one-way hash chain structure prevents replay attacks, while the CGA prevents falsified binding updates. The scheme provably meets its security goals under the well-established Dolev-Yao threat model used in AVISPA. Thus, through formal methods, cryptographic constructions, and systematic analysis, the paper provides a robust security argument covering a broad range of real-world attacks applicable to binding update protocols within the standard model.

5. Performance evaluation

This section discusses the performance evaluation of the One-CLU scheme in terms of signaling cost, binding update cost, and packet arrival rate.

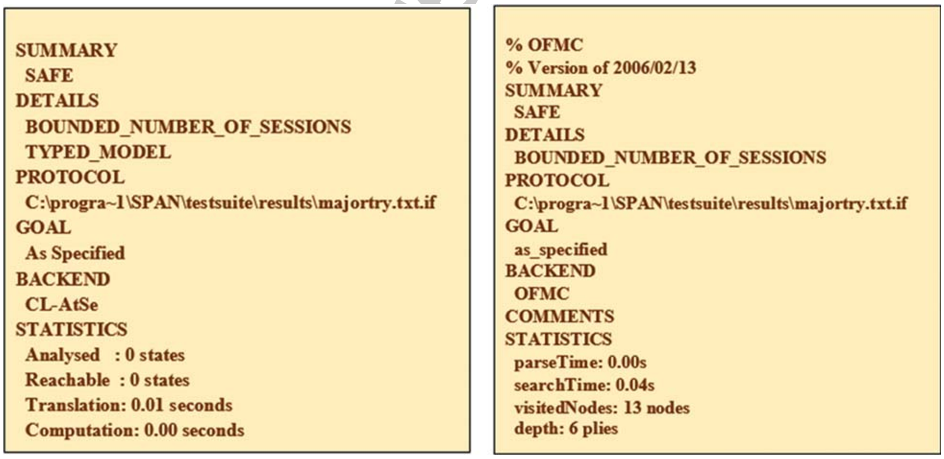
5.1. Simulation set-up

The simulation set-up of the One-CLU scheme is implemented in OMNeT++. It is an object-oriented modular network simulation framework with a component architecture that allows it to be widely used in various domains [19]. It uses a high-level language known as network description to assemble it into a larger component and describe its structure. The MIPv6 allows an MN to maintain the connections transparently while moving from one network to another. The proposed One-CLU scheme's experimental set-up consists of MN, CN, HA, and access routers, as shown in Fig. 5.

5.2. Numerical results

The results are analyzed in the INET 2.9 framework designed for the MIPv6 environment in the OMNeT++ simulator [20]. In MIPv6, the transmission time of each BU message between MN and HA is reduced when the MN resides in its network (home network). Typically, when the MN is in a visited network, there is a delay in packet transmission to the recipient due to wireless links. Hence, it increases transmission time with a larger packet size, as shown in Fig. 6.

Binding refresh (BR) refers to updating old BU messages when new messages arrive in the cache [21]. The cost of binding refresh increases when the BU messages in the cache are frequently updated [22]. Hence, the cost for BR is measured to be a limitation associated with existing schemes. It is addressed in the proposed method, which reduces the frequency of BU messages from MN to the CN and HA. The cost of BR for our One-CLU scheme is computed as follows: $\text{Cost}_{\text{BR}} = 2\text{M}(\text{BR}_{\text{HA}}) + 2\text{M}(\text{BR}_{\text{CN}})$, where $M(BR_{HA})$ is the mean value of the number of BR messages sent to HA and $\text{M}(\text{BR}_{\text{CN}})$ is the mean value of the number of BR messages sent to CN. Accordingly, the cost of BR for the proposed scheme is estimated as $254.94335 + 509.88670 = 764.830053$ per unit time. The cost analysis of BR for various schemes with varying lifetimes of BU is shown in



```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 C:\progra~1\SPAN\testsuite\results\majortry.txt.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
 Analysed   : 0 states
 Reachable  : 0 states
 Translation: 0.01 seconds
 Computation: 0.00 seconds
```

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 C:\progra~1\SPAN\testsuite\results\majortry.txt.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.04s
 visitedNodes: 13 nodes
 depth: 6 plies
```

Fig. 4. OFMC and CL-AtSe results of the One-CLU scheme.

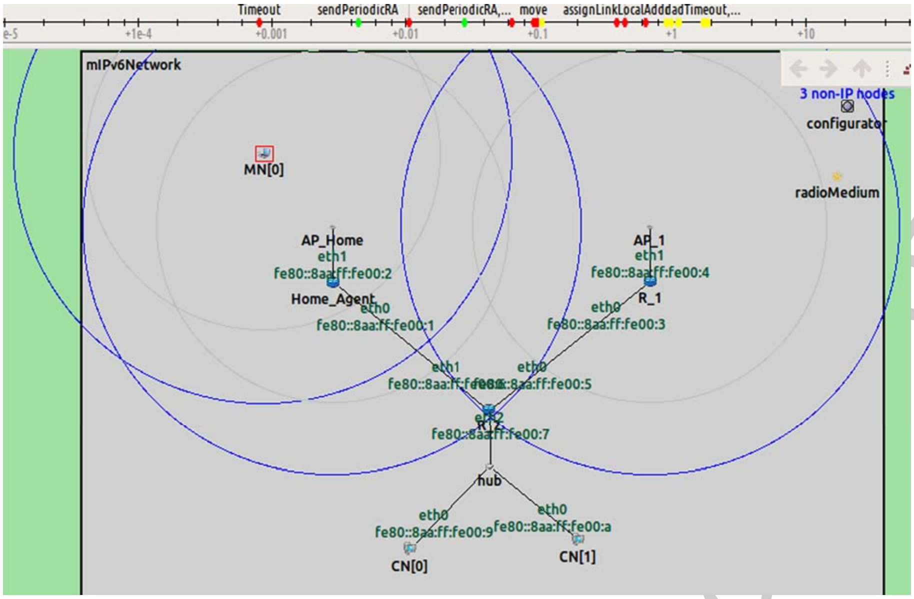


Fig. 5. Environmental set-up in OMNeT++.

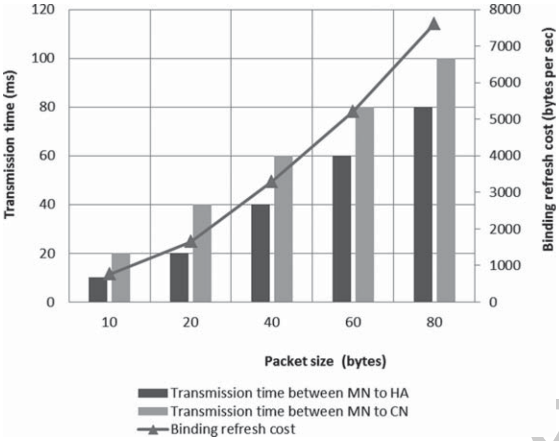


Fig. 6. Transmission time vs. binding refresh.

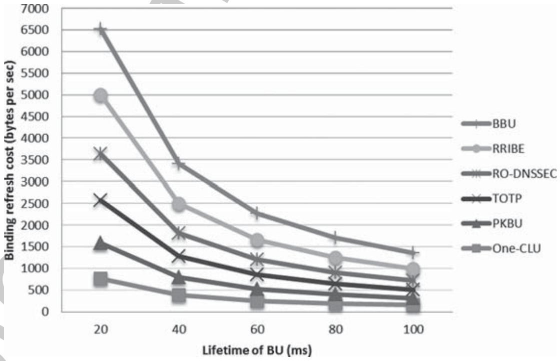


Fig. 7. Binding refresh cost vs. lifetime of BU.

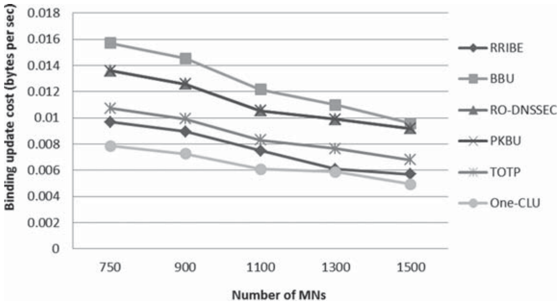


Fig. 8. BU cost with an increase in the number of MNs.

Fig. 7. In a MIPv6 network with varying sizes, the BU cost from MN to HA and MN to CN varies with the increase in mobile nodes, as shown in Fig. 8. The proposed One-CLU scheme holds lower BU cost since the number of local binding updates (LBU) and global binding updates (GBU) to HA and CN are reduced compared to existing schemes. The BU cost for the proposed One-CLU scheme is calculated as follows,

$\mathrm{Cost}_{\mathrm{BU}}$ = Number of hops that the MN stays within the access network × $\mathrm{Cost}_{\mathrm{LBU}}$ + number of BU's when the access network crosses its domain × $\mathrm{Cost}_{\mathrm{GBU}} = 0.0053$ per unit time, whereas the $\mathrm{Cost}_{\mathrm{BU}}$ of BBU is 0.0122 per unit time. The above result shows that the proposed scheme outperforms BBU since its $\mathrm{Cost}_{\mathrm{BU}}$ is less than BBU. Increasing the packet delivery cost ($\mathrm{Cost}_{\mathrm{Packet\_delivery}}$) significantly

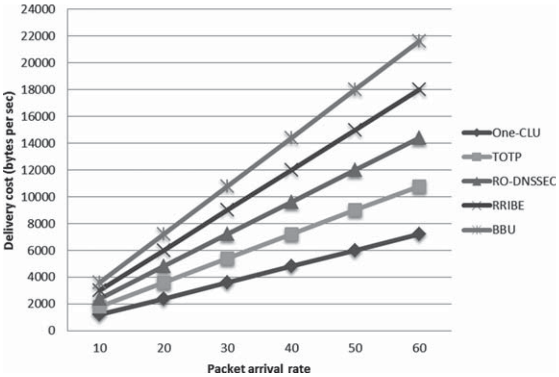


Fig. 9. Packet delivery cost vs. arrival rate.

reduces the overall effectiveness of the BU scheme due to packet loss during transmission.

The delivery cost for the total number of packets sent to HA and CN is shown in Fig. 9. The proposed One-CLU scheme eliminates the redundant signaling messages to increase overall efficiency. Here, the packet delivery cost is estimated as follows,

$$\mathrm{Cost}_{\mathrm{Packet\_delivery}} = \mathrm{k} \times \mathrm{Cost}_{\mathrm{tunnelling}}$$

$$+\mathrm{s} \times \mathrm{Cost}_{\mathrm{Packet\_loss}}$$

where k and s are constants and $k + \mathrm{s} = 1$.

The $\mathrm{Cost}_{\mathrm{Packet\_delivery}}$ of the proposed scheme is estimated as 0.0022 per unit time, and it is lower than TOTP with 0.0153 per unit time, which is a lower cost than RO-DNSSEC with 0.0317 per unit time.

The handover issue greatly impacts the BU scheme as the mobile nodes frequently change locations [23–25]. As it undergoes more handovers, packet loss's probability increases, as shown in Fig. 10. The packet loss in the proposed One-CLU scheme is drastically reduced by eradicating the BUs in the communication. Finally, the total signaling cost per unit time for the BU scheme is computed as,

$\mathrm{Cost}_{\mathrm{Total\_signal}}$ = number of hops that MN stays within the access network × $\mathrm{Cost}_{\mathrm{LBU}}$ + the number of BU's when the access network crosses its domain × $\mathrm{Cost}_{\mathrm{GBU}}$ + 2 × average number of BR messages sent to HA + 2 × average number of BR messages sent to CN + $\mathrm{Cost}_{\mathrm{Packet\_delivery}}$.

Consequently, the total signaling cost for the proposed BU scheme is calculated as $0.0053 + 764.8300 + 0.0022 = 764.8375$ per unit time. The results reveal that the proposed scheme costs less than the existing schemes. The performance of each of these schemes for BU cost is depicted in Fig. 11.

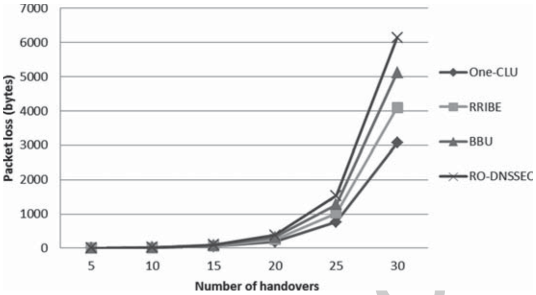


Fig. 10. Comparison of packet loss with the number of handovers.

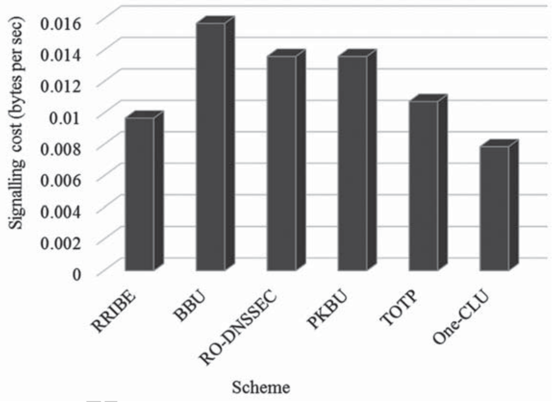


Fig. 11. Comparison of total signaling cost.

The effectiveness of the proposed One-CLU BU scheme is demonstrated in terms of reducing communication payloads and costs for binding update, binding refresh, and packet delivery. The simulation results show that the proposed scheme reduces the transmission time of each binding update message between the mobile node and home agent and reduces the cost of a binding refresh. The proposed scheme achieves this by generating a unique CoA for each network, which is cryptographically generated using the one-key-based CGA technique. Additionally, the proposed scheme conceals the identity of the mobile node by using a temporary CoA instead of the permanent HoA. The scheme exhibits resilience to handover issues, evident in a significant reduction in packet loss probability, showcasing real-world effectiveness. Overall, the comprehensive evaluation confirms the One-CLU scheme's cost-effectiveness compared to existing methods, as highlighted in graphical representations. This research contributes significantly to advancing mobility management, offering a robust solution for MIPv6 networks.

The key assumptions and limitations are acknowledged to provide a comprehensive understanding of our proposed BU scheme. Firstly, the efficacy of One-CLU relies on the presumption of a trusted infrastructure encompassing both home and visited networks. The cryptographic strength of the one-key-based cryptographically generated address (CGA) is a foundational assumption, making the security of the scheme contingent upon the robustness of cryptographic primitives. Additionally, the scheme assumes user compliance with established security practices and the absence of activities that might compromise the binding update process or divulge sensitive information. Moreover, continuous network availability for both home and visited networks is assumed for timely location updates. Moving to limitations, scalability to extremely large networks requires further analysis. Interoperability issues may arise in heterogeneous network environments, and resource constraints on mobile devices, such as limited processing power, could affect performance. Furthermore, dynamic network conditions and potential vulnerabilities to advanced attacks are acknowledged as limitations, highlighting the need for further exploration and refinement in real-world scenarios.

6. Conclusion

The paper proposes a new secure and efficient One-CLU scheme for MIPv6 by integrating a one-key-based CGA method. Here, the proposed scheme provides mutual authentication between the communicants, and in addition, the CN verifies the CoA to ensure that MN is a valid node. It significantly reduces security vulnerabilities as the participants compute their secret key independently without pre-sharing it explicitly. It avoids additional signaling costs by limiting the number of BUs between communicants, thereby reducing BU's handover latency. The security features of the proposed BU are validated using AVISPA, ensuring that it is safe and does not contain any security flaws for attacks such as MITM, DoS, false binding update, session hijacking, amplification, and replay. The performance evaluation shows that our One-CLU scheme's efficacy is significantly upgraded than existing BU schemes. While this proposed BU scheme shows promise, further extensions of this work should focus on real-world testing and large-scale deployment of the binding update scheme. The scheme could be implemented in a network with more devices to evaluate performance under dynamic conditions for scalability. An additional focus on refining and optimizing the scheme's resilience in varying network scenarios would be beneficial. Moving from theoretical design to practical implementation and stress testing is needed to strengthen this scheme as a robust security solution for next-generation mobile networks.

References

[1] S. Nam and S.G. Min, An identifier locator separation protocol for the shared prefix model over IEEE WAVE IPv6 networks, *IEICE Transactions on Communications* **106**(4) (2023), 317–330.

[2] N. Dutta and H.K.D. Sarma, Efficient mobility management in IP networks through three layered MIPv6, *Journal of Ambient Intelligence and Humanized Computing* (2021), 1–19.

[3] K. Pokhrel, N. Dutta M.K. Ghose and H.K.D. Sarma, Performance evaluation of three layer MIPv6 architecture. *Wireless Personal Communications* **128**(2) (2023), 1259–1285.

[4] S. Mathi, E. Joseph, M.S. Advaith, K.S. Gopikrishna and R. Gopakumar, A flattened architecture for distributed mobility management in IPv6 networks, *Journal of Intelligent and Fuzzy Systems* **38**(5) (2020), 6583–6593.

[5] S. Ibrahim and Y. Mohamed, A model for enhancing nested mobile nodes performance, *Proceedings of the International Conference in Advances in Power, Signal, and Information Technology IEEE*, 2023, 411–418.

[6] S. Arvind and V.A. Narayanan, An overview of security in CoAP: attack and analysis, *Proceedings of the 5th International Conference on Advanced Computing and Communication Systems, IEEE*, 2019, 655–660.

[7] A. Pillai, M. Sindhu and K.V. Lakshmy, Securing firmware in Internet of Things using blockchain, *Proceedings of the 5th International Conference on Advanced Computing and Communication Systems, IEEE*, 2019, 329–334.

[8] W.A.A. Alsalihy and M.S.S Alsayi, Integrating identity-based encryption in the return routability protocol to enhance signal security in mobile IPv6, *Wireless Personal Communications* **68**(3) (2013), 655–669.

[9] V.R. Reddicherla, U. Rawat, Y. Kumar and A. Zaguia, Secure vertical handover to NEMO using hybrid cryptosystem, *Security and Communication Networks*, 2021.

[10] A. Rossi, S. Pierre and S. Krishnan, Secure route optimization for MIPv6 using enhanced CGA and DNSSEC, *Systems Journal, IEEE* **7**(3) (2013), 351–362.

[11] S. Nowaczewski and W. Mazurczyk, Securing future internet and 5G using customer edge switching using DNSCrypt and DNSSEC, *J Wirel Mob Networks Ubiquitous Comput Dependable Appl* **11**(3) (2020), 87–106.

[12] S.S. Gosavi and G.K. Shyam, A novel approach of OTP generation using time-based OTP and randomization techniques, In *Data Science and Security*, Springer, 2021, 159–167.

[13] C.S. Park and H.M. Nam, A new approach to constructing decentralized identifier for secure and flexible key rotation, *IEEE Internet of Things Journal*, 2021.

[14] L. Zhang, M. Ma and Y. Qiu, An enhanced handover authentication solution for 6LoWPAN networks, *Computers and Security* **109** (2021), 102373.

[15] S. Chandrasekaran, K.I. Ramachandran, S. Adarsh and A.K. Puranik, Avoidance of replay attack in CAN protocol using authenticated encryption, *Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies, IEEE*, 2020, 1–6.

[16] M. Nakkeeran and S. Mathi, A generalized comprehensive security architecture framework for IoT applications against cyber-attacks, In *Artificial Intelligence and Technologies*, Springer, Singapore, 2022, 455–471.

[17] L. Babenko and I. Pisarev, Translation of cryptographic protocols description from Alice-Bob format to CAS+ specification language, *Proceedings of the International Conference on Intelligent Information Technologies for Industry*, 2019, 309–318.

[18] P.R. Yogesh, Formal verification of secure evidence collection protocol using BAN logic and AVISPA, *Procedia Computer Science* **167** (2020), 1334–1344.

[19] P.A.B. Bautista, L.F. Urquiza-Aguiar, L.L. Cárdenas and M.A. Igartua, Large-scale simulations manager tool for OmNet++: Expediting simulations and post-processing analysis, *IEEE Access* **8** (2020), 159291–159306.

[20] K. Kuladinithi, R. Elsner, L. Krüger, S. Lindner, C. Petersen, D. Plöger and A. Timm-Giel, Teaching modelling and analysis of communication networks using OMNeT++ simulator. In *OMNeT++*, 2018, 111–123.

[21] N. Dutta and H.K.D. Sarma, Efficient mobility management in IP networks through three-layered MIPv6, *Journal of Ambient Intelligence and Humanized Computing* 2021, 1–19.

[22] K. Pokhrel, N. Dutta, M.K. Ghose, H. Vithalani, H.K. Sarma and Z. Polkowski, Binding lifetime based signaling cost analysis of multilayer MIPv6, *Journal of Computers* **13**(3) (2018), 337–350.

[23] S.M. Ghaleb, S. Subramaniam, Z.A. Zukarnain, A. Muhammed and M. Ghaleb, An efficient resource utilization scheme within PMIPv6 protocol for urban vehicular networks. *PloS One* **14**(3) (2019), e0212490.

[24] S. Pandey, G. Kadambi and V. Pande, Applying bipartite supergraphs to mitigate ping pong effect in hierarchical wireless networks, *Proceedings of the Third International Conference on Advances in Electronics, Computers and Communications, IEEE*, 2020, 1–6.

[25] P. Sapkale, U.D. Kolekar and N. Kumar, Mobility management based mode selection for the next generation network, *Proceedings of the International Conference on Ubiquitous Communications and Network Computing*, Springer, 2021, 16–25.