Received: 24 April 2023 | Revised: 19 Oct 2023 | Accepted: 20 Dec 2023 | Published: 30 Dec 2023
Volume 1(2) Pages 71-92
DOI: https://doi.org/10.56279/jicts.v1i2.32

JICTS
Journal of ICT Systems

# Security Enhancement of SIP Protocol in VoIP Communication

**Hussein Sudi Lema[1], Fatuma Simba, Joseph Cosmas Mushi**

*Department of Computer Science and Engineering, University of Dar es Salaam, Dar es Salaam, Tanzania*

[1]Corresponding author
Email: husssul@gmail.com

**Funding information**

This work is a result of PhD research work. It was full funded by the University of Dar es Salaam.

**Abstract**

In Voice over IP (VoIP) systems, calls are started, maintained, and ended using the Session Initiation Protocol (SIP). One of the SIP flaws is the lack of enough data to validate users, and its contents can be changed to fake the Caller Identification (CID). Fake CID can be used by hackers to trick receivers and obtain valuable materials. Existing solutions have a number of faults, including the use of more network resources, the use of insufficient data to identify fake CID, and high call setup delay for caller and callee during the validation process. This work presents an algorithm to enhance SIP protocol security to mitigate the flaws of the previous solutions that are used to address CID spoofing. By using the Media Access Control (MAC) address, the algorithm can validate the CID and warn the callee for the fake CID. The algorithm was developed and tested by using Mininet, Python based open source OpenFlow (POX) controller, SIPp simulator, Linphone softphone, and freePBX. All faked CID were detected and receivers were alerted on the linphone screen. Session setup delay obtained are within the 15.1 ms to 27.3 ms, which are within the acceptable Quality of Service (QoS) ranges.

***Keywords***

*Caller identification*

*MAC Address*

*Session setup delay*

*SIP protocol*

*VoIP Systems*

## 1. Introduction

Voice over Internet Protocol (VoIP) facilitates transmission of voice and video signals instantly over Internet Protocol (IP) [1]. It has a low cost of building infrastructure and low operation charges. Therefore, in terms of service charges and infrastructure implementation, it enables video/audio calls over a longer distance to be cheaper than traditional communication [2].

VoIP communication uses Session Initiation Protocol (SIP) for initiating, maintaining, and ending audio/video calls [3]. However, SIP is vulnerable because it sends plain text messages. Hackers can access and change information present in SIP headers [4]. Also, it lacks the data necessary to verify a caller [5]. As a result, by exploiting SIP information, a caller/hacker may spoof Caller Identification (CID) without being noticed.

SIP Authentication is a process where users are authenticated with a VoIP Network and gain access to the system for the use of VoIP Services [6]. The SIP Authentication process is done with the User Datagram Protocol (UDP) packets from the user to the VoIP Server, and needs USERNAME and PASSWORD [7]. Most VoIP machines do not use any Encryption techniques to authenticate users. This kind of vulnerability found in most VoIP services is also known as “insecure communication vulnerability” [1]. Therefore, attackers can use some third-party applications/tools to spoof CID. Some applications are SVmap, Svwar, InviteFlood, SIPDump, Armitage and Metasploit (SIP_invite_Flood) [1].

The CID spoofing attack is simple to carry out but challenging to defend against [8]. Also, CID spoofing may be used by several attacks to disguise the attacker’s identity. It convinces the call recipient to believe the attacker/caller is trustworthy [6]. The outcomes of the fake CID name are identified after the communication. It may cause significant loss of valuable material and/or the leaking of private information [9]. It was difficult to identify fake CID because there is no standard unique identification, such as a MAC address on a LAN network to identify CID.

The MAC address is the network protocol used to transfer packets by using layer 2 switches within the Local Area Network (LAN). It is used to identify local network devices, which are unique to each device [10]. Also, a MAC address allows users to get access to internet services on the LAN after being validated [11]. In addition, it is used to detect evil twin attacks [12]. However, the MAC address can be hacked by spoofing its value [13], but it can be mitigated by various mechanisms [13]. Most of today’s network security solutions are implemented by using Software Defined Networks (SDN) [14].

SDN is a technology that was developed to enhance network security [15]. It separates the data plane from the control plane [14]. The data plane is used to transfer packets within the OpenFlow switches, while the control plane is used to control the flow of packets in a centralized controller [16] [15]. A controller is used to control the transfer of packets. By setting up and implementing policies and algorithms by using programming languages, it enables control of the flow of packets [17]. Most of the security mechanisms are developed by using SDN technology, which makes the solution simple and easy to implement [18]. For example, a solution to prevent IP spoofing attacks [18].

Several solutions have been proposed to solve CID spoofing. They are of two categorized, which are end-to-end solutions and network solutions. End-to-end solutions include the Callee-only inference and verification (CEIVE) application [19], Caller ID Verification (CIV) [20], and CallerDec application [5], [21]. They use the challenge-and-response method, whereby a recipient sends a verification call/message back to the caller and waits for a response from the caller. Therefore, communication between a caller and a callee causes the utilization of more shared network resources, causing a Denial of Service attack [5].

In the network solutions category, which are the iVisher system [22] and Network ASsisted CID authentication (NASCENT) system [9], these mechanisms need GateWay(GW) to be upgraded to enable communication between GW and the IP Multimedia Subsystem (IMS) or Private Branch eXchange (PBX) [22], [9]. The upgraded GW needs to be programmable to insert a code that can analyze the packets and return new calls/messages to PBX/IMS.

This study aimed to address shortcomings of the previous existing solutions for CID spoofing attacks. The study aimed to enhance SIP protocol security by adding a MAC address to authenticate callers. We designed, implemented, and tested an algorithm that addressed the solution of CID spoofing attacks. Through the POX controller and PBX in the dialplan, the CID name was validated. The performance of VoIP services was evaluated using the implemented algorithm.

Therefore, the proposed solution can validate the CID name with a minimum shared network resource. By reducing intercommunication with extra new calls or messages, the proposed solution uses original SIP messages to shorten call setup delays caused by validation processes. The proposed solution can be deployed easily because it needs only software, with no additional devices or upgrades of existing devices. A solution is targeted to be piloted within an organization, with information records for its customers. Therefore, its MAC and IP address of users can be easily stored.

## 2. Related Works

Several solutions have been proposed in previous studies to address CID spoofing. They include challenge-response solutions and network-assisted solutions. Some of the challenge-response solutions are CEIVE [19], CallerDec [5], and CIV [20]. They involved intercommunication between the caller and the callee by using a Short Message Service (SMS) [5] or a call [17]. The caller's response is required for the solution to work. In addition, these solutions increase call setup delays by 8.35 seconds to 20 seconds. By using these solutions, a caller/hacker can conduct a DoS attack on a callee/victim [19]. Hackers can make several calls with fake CID on which a new call or SMS that is used to validate CID use more shared network resources, causing a DoS attack [23], [4], [19].

The CIV authenticates the caller ID based on a challenge-response process instead of digital signatures, hence requiring no PKI [20]. It works similar to the CallerDec application and the CEIVE application. The CIV requires support from both calling participants. Also, CIV uses the Dual-Tone Multi-Frequency (DTMF). It implements the challenge-response process on VoIP, cellular and landline phones. The solution can work across heterogeneous networks (Signaling System (SS7)/SIP). It works by only updating the software on the user's phone. Therefore, CIV has the same effect as CEIVE and CallerDec applications.

With assisted network solutions, the callee can communicate with a GW, and the GW can communicate with PBX [22] and/or with the IP Multimedia Subsystem (IMS) [9]. These solutions have several flaws. Firstly, they use information that is not available to some devices on VoIP communication. For example, Mobile Station International Subscriber Directory Number (MSISDN) and or International Mobile Subscriber Identity *(*IMSI) are used to validate CID on NASCENT [24]. This information is found in User Equipment (UE) on the Subscriber Identity Module (SIM) Cards [9]. Secondly, there is intercommunication between IMS and Evolved Packet Core (EPC) during the mapping process, which uses more shared network resources [9], and between Gateway and PBX [22]. In addition, the Packet Data Network Gateway (PGW) must upgrade its hardware so that it may install software to communicate with IMS or PBX [22]. Finally, they increase the call setup delay. The call setup delay in a normal call is approximately 0.715 seconds; using a validation system, it takes 0.994 seconds [22].

Another existing solution uses a MAC address, IP address and SIP Uniform Resource Identifier (URI) as a device profile for the caller side [11]. This solution removes the privacy of the users because they can end the call when the caller identification is incorrect. Also, when the INVITE request message reaches the callee side while CID was suspected to be spoofed, a Callee should ask a SIP proxy to verify. Therefore, it adds an intercommunication between Callee and the SIP proxy, causing call setup delays to be high. It provides an additional delay of about 27.125 ms [11] more than the normal call without verification.

Also, there is a Secure Telephone Identity Revisited (STIR)/Signature-based Handling of Asserted Information Using toKENs (SHAKEN) solution. It is used to protect identity by encrypting SIP messages with a public key. Then, the carrier identity can be authenticated by the first Internet Telephony Service Provider (ITSP) and then authenticated by the next ITSP [25]. STIR/SHAKEN does not authenticate any caller ID. Instead, it authenticates the original/first gateway carrier of the caller. Also, it can authenticate the gateway carrier for international calls that arrive inbound at the gateway [20]. Also, because STIR/SHAKEN relies on a public key infrastructure (PKI) to manage digital certificates, scaling up this PKI for the global telecom industry is rather difficult [20]. Most solutions that use encryption techniques degrade the Quality of Service (QoS) into the VoIP system [22, 25, 26, 27, 29].

The study solution was made on a LAN network by using only the first ITSP or PBX within the LAN and outside communication can be supported with any encrypted mechanism to hide/protect hackers from spoofing MAC or CID name. MAC address on the controller storage can be validated by frequency check up with ARP protocol and manually verified by the system administrator. After validation, the MAC address was removed to mitigate misuse of MAC address by hackers/recipients of the call.

## 3. Scope of the Proposed Solution

Before making a call, the proposed solution used the Address Resolution Protocol (ARP) to collect the MAC and IP addresses of the LAN users. Then, the MAC and IP address were stored in the Controller database within the LAN. Currently, these data are stored in comma-separated values (CSV) files to be easier to extract by using the SIPp simulator codes. Sample data stored in a POX controller in CSV file are found in Appendix F. During the call setup, the stored MAC address is used to compare with that present in the packet (INVITE request message) to validate the MAC address within LAN in POX controller. Therefore, there is no outside users can use an organization LAN for call.

The proposed solution involves an algorithm to enhance the SIP protocol by adding the MAC address to an INVITE and REGISTER request message. REGISTER request messages used to insert MAC, CID name and number to the asterisk database. Currently, these details are inserted manually.

An INVITE request message extracts the MAC address after being validated with IP addresses on the controller. Then, the embedded MAC address, extension and CID name are validated within the asterisk server on the dialplan. Sample codes of the CID validation are found in Appendix E.

The proposed solution used SIP messages that are used to start calls to minimize latency and utilization of more shared network resources. The results are collected by using trace_error and trace_msg of the SIPp simulator log files. Also, Wireshark and Linphone screens were used to collect data that detect and warn CID spoofing attackers to a callee.

The current study did not work with an intermediate attack, such as a Man in the Middle. Existing Man in the Middle solutions can, however, be used with the proposed solution. We suggest using any encryption, such as Transport Layer Security (TLS) and Internet Protocol security (IPsec), to prevent middle hackers during communication [28].

Also, the authors of this study did not develop REGISTER request messages that were used to insert MAC and IP addresses into the astDB. However, the proposed solution inserted MAC addresses, IP addresses, CID names and numbers manually by using the SQLite browser.

MAC and IP address spoofing were not considered in our study. However, it may be prevented by using another solution that was used to validate the IP address with the MAC address [12]. This validation was conducted in the controller.

We worked on analyzing packet size before and after adding the MAC address to the SIP messages, which tests how validation uses bandwidths. Also, we worked on the Session Setup Delay (SSD) during the call setup, which analyzes the latency of the call setup before and after inserting the validation mechanism. Finally, our study involved detection of caller ID spoofing on the Wireshark and Linphone screens to show if CID is spoofing or not. The proposed solution provides warning to the victim.

The study assumes that the solution should be implemented within organizations. Organizations, such as banks and government agencies that provide potential information or services to customers, may use the proposed solution. Any unauthorized users are restricted from accessing an organization’s network. The aim was to prevent unauthorized users from accessing information within the organizations. Therefore, hackers cannot get the MAC address of the devices within the organization. They can get by trapping packets in the middle of communication that can be protected by encrypting the data by using other encryption mechanisms. However, the organization can receive a call from customers with any CID (not validated incoming calls).

Any man-in-the-middle attackers who can capture the MAC and IP address of the caller within the organization or elsewhere cannot assist with any encrypted mechanism. They can make a call and be validated as a correct caller. But they need to spoof four fields: IP, MAC addresses, CID name, and CID number.

## 4. Design of the Proposed Solution

An algorithm was designed and implemented in the POX controller to collect and store IP and MAC addresses within the LAN. Also, it validates the CID name within the PBX on LAN or Caller service provider. During the configuration of the network devices, Layer 3 Switch (L3S) used the ARP protocol to collect and store MAC and IP addresses within the POX controller. The MAC address and IP address were stored after checking if they were valid [12] and whether the MAC address exists or does not exist in a database.

When the User Agent Client (UAC) registers with the Registrar server, the Controller inserts the MAC address in the REGISTER request message. Then, REGISTER messages are sent to the SIP server for storage in the Asterisk database. Currently, this process was done manually.

When the call setup is made, the INVITE SIP request message takes the MAC address from the Controller in the CSV file to the PBX to validate CID. The algorithm implemented to the dialplan in the extensions_override_custom.conf file. Both the astDB and the INVITE request message retrieve the MAC address, CID number, and CID name from their storage. Finally, the comparison is made by using data in the INVITE request message with that on the astDB.

Before adding the MAC address to the INVITE message, IP and MAC addresses are verified [12]. If they are fake, the INVITE message is dropped and a new INVITE message waits for another call setup. It inserts the MAC address value into the SIP INVITE message if the MAC address is valid. Finally, it sends the packet to the PBX for CID validation. These steps are shown in Appendix I.

To generate the INVITE/REGISTER SIP request message, we used the SIPp simulator. Appendix H shows the flowchart of the steps that validate the CID name. Steps/procedures used to validate CID are as follows:

i. Wait for a packet.
ii. Check if the packet has an INVITE request SIP message.
iii. If it is not an INVITE request message, the process leaves the packet to proceed according to its request and goes on to wait for another packet.
iv. If it is an INVITE request message, the context code extracts Display_name, extension from the "from" header field, and MAC address value from the SIP header.
v. Also, the context code extracts the CID name, CID number, and MAC address from the asterisk database (astDB).
vi. Then, compare the MAC address from the INVITE request message with the MAC address on the astDB database.
vii. If the MAC address is not the same or not present, assign the CID name to the display name with the word "Fake_MAC_ADDR". Then, the context allows the dial method to call the receiver and finally hang up a call after communication.
viii. If the MAC address is the same, compare the CID name and number from the INVITE request message with those in the astDB.
ix. If the CID name and number are the same, assign the CID name to the display name (CID name from the INVITE request SIP message). Then, the context allows the caller to dial the call to the receiver and finally hang up a call after communication.
x. If they are not the same, assign the CID name to the word "fake" plus the display name. Then, context allows dialing the call to the receiver (on which its display name will show CID name is fake) and finally hangs up a call after communication.
xi. Finally, after communication is completed, any UAC can end a call by hanging up the call. Then, it waits for another INVITE packet is repeated.

The solution ensures that other users cannot access or manipulate IP and MAC addresses by storing them in the Database. Before adding the MAC address to the INVITE request message during the call setup, IP and MAC addresses are checked to ensure that they are valid. Also, it protects consumers from CID spoofing by validating the CID using the MAC address encoded in the INVITE message within the PBX dialplan.

However, the proposed solution cannot be protected by the Man in the Middle attackers. An attacker can steal information within the packet and use it later or inject new information to redirect the communication. The study suggested using a different mechanism, such as Transport Layer Security (TLS) [28], to defend against a Man in the Middle attack.

For the suggested solution to function, extra storage devices and software (POX controller) are needed. IP and MAC addresses are validated and stored in a database/CSV file using codes inserted in the L3S on l3_learning.py. Other codes are implemented within the PBX. Therefore, most of the additional tools/infrastructure are deployable to most of the systems.

## 5. Experimental Setup

Figure 1 shows connected devices on the experimental setup. The algorithm is tested using three smartphones [8], two laptops, and a TP-Link access point.

The proposed solution used Design Science Research Method (DSRM) [29]. DSRM allows cycling forwarding to the next steps or/and backward to earlier/previous activities. For example, one can move back from the Evaluation step or communication step to the Define Objectives step or Design and Development step. Table 1 shows the steps proposed by the DSRM methodology to develop the proposed solution. Appendix A shows a summary of the DSRM steps used to develop the proposed solution.

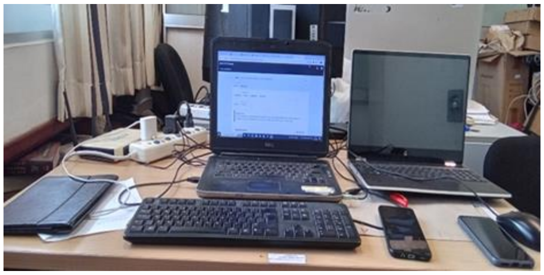


Figure 1. Equipment setup.

The first laptop has 8 GB of RAM, Core (TM) i7-3520M CPU @ 4.0GHz speed, and a hard disk of 1.0TB storage. This laptop was installed with the following software: Wireshark, Zotero/Juris-M, DB Browser for SQLite, SIPp simulator, Mininet emulator, and Linphone softphone tools. When making calls on this laptop, both a regular user and a virtual user can be used. Also, it serves as a tool for gathering data.

The second laptop has 16 GB of RAM, Core (TM) i7-3520M CPU @ 8.0GHz processor speed and 1.0TB hard disk storage. It was installed with FreePBX as a VoIP server. FreePBX dashboard can analyze the performance of the asterisk server in terms of usage of the Central Processing Unit (CPU) and memory. Also, it is used as an asterisk server to enable communication by registering a UAC and managing the calls.

Three Android smartphones were used as UACs. The Linphone softphone was configured as a calling application for smartphones. They were used for making calls directly to the Mininet host without passing through the controller and to accept calls from Mininet hosts. Every UAC device was configured to answer every SIP call automatically.

Using the Mininet emulator and Python code, we developed a network topology that uses SDN [30]. Figure 1 shows such a topology. Host within the Mininet communicated with other devices by using the SIPp simulator with various MAC, and CID names and numbers. OpenFlow switch was connected to the network by using the network port on the laptop (enp2s0). Then, the network port was connected to the internet through an Ethernet cable to the wireless access point on TP-link. By using codes in Table 1, we built network topology with a POX controller (Figure 2). A complete sample code is found in Appendix B.

Table 1. Sample Code to build a network topology within the Mininet emulator.

| Sample Code to create virtual network topology |
|---|
| net.addController(name=’c0’)<br>s1 = net.addSwitch(’s1’)<br>Intf( ‘enp2s0’, node=s1 )<br>net.addLink(h1, s1)<br>h1 = net.addHost(’h1’, ip=’0.0.0.0’) |

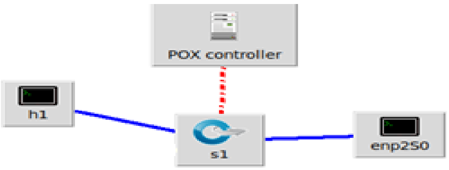


Figure *2*. SDN topology made by using Python code within the Laptop on the Mininet.

The testbed experiment is conducted on a Linux Operating System. The aim was to use different Linux commands and applications. SIPp simulator was used to generate SIP messages with and without fake CID names. Also, it can add MAC addresses on INVITE request messages from the CSV file. The SIPp simulator can generate and send several types of SIP packets with several numbers of packets. The Mininet tool was used to create network topology with two hosts connected to L3S. Then, L3S was connected to the POX controller.

L3S was connected to the internet through the Ethernet cable on the TP-link. The three smartphones and the FreePBX communicate with the host in Mininet through a TP-link router with wireless access.

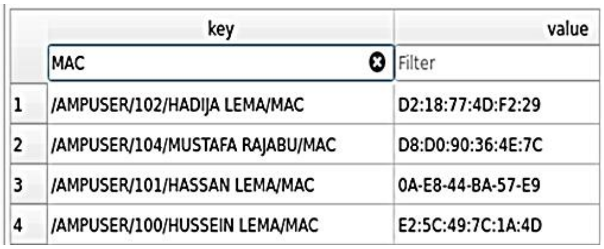
Wireshark, POX controller and SIPp simulator were used to capture the results of the proposed solution. They captured transmitted packet information and errors that occurred during communication. POX and SIPp log files are used to identify the size of the SIP messages, spoofed CID, and errors/warnings that exist during the running of the algorithm. Also, they were used to analyze the performance of the SIP packets, such as failure and success of packet processing. Wireshark was used to capture packets on the Mininet host through the SIPp command. Also, it can capture packets from the smartphone that have made a call to the laptop. The data captured from Wireshark were collected in the spreadsheet file; and they are used to calculating the Session set-up delay (SSD) during the call setup.

Our study used the Berkeley Database to store the MAC address, CID number and name in the astDB. Berkeley Database is an open source, with several key advantages over the relational database. It is simple to use and supports concurrent access by multiple users. It provides transaction support such as the surviving system and disk crashes [21].

POX controller is used to access IP and MAC addresses from LAN users and store them in the CSV file. The flowchart that shows steps of collecting and storing IP and MAC address are found in Appendix I. The steps used to collect and store IP and MAC address are as follows:

i. Wait for ARP packets.
ii. Check for the packets that arrive at the controller if they are Address Resolution Protocol (ARP) packets or not.
iii. If it is an ARP reply packet, check if the IP and MAC address are present in the packet, and if they are present on the database. In our case, we use a CSV file.
iv. If the IP and MAC address are present in the database, the packet is sent to the L3S switch for the next steps.
v. If it is absent, it is added to the database on the controller and then forwarded to the L3S switch for the next steps.
vi. If the packet is not an ARP packet, the controller checks if the packet is an INVITE/REGISTER request SIP message.
vii. If it is an INVITE/REGISTER request SIP message, extract the IP address from SIP message and compare it with the IP address in the database.
viii. If an IP address is valid, the MAC address is added to the SIP header and forwarded from the L3S switch to the PBX.
ix. If an IP address is not valid, the packet is discarded/dropped and waiting for another packet.
x. If the packet is not INVITE/REGISTER request SIP message, the packet is forwarded to the L3S switch to be processed as required by that type of packet.
xi. After completing all steps and the packet forwarded, the process was repeated to wait for another packet, and the process/steps were repeated.

Sample codes that collect and store IP and MAC addresses are found in Appendix C. The stored IP and MAC addresses allow the SIPp simulator to read the MAC address value from database (CSV file). SIPp simulator scenarios use word MAC as a new SIP header field into the INVITE/REGISTER request SIP messages to insert MAC address value. Sample codes to create SIP messages with MAC address are found in Appendix D. The data written into the astDB is shown by using DB Browser for SQLite (Figure 3).



| | key | value |
|---|---|---|
| | MAC | Filter |
| 1 | /AMPUSER/102/HADIJA LEMA/MAC | D2:18:77:4D:F2:29 |
| 2 | /AMPUSER/104/MUSTAFA RAJABU/MAC | D8:D0:90:36:4E:7C |
| 3 | /AMPUSER/101/HASSAN LEMA/MAC | 0A-E8-44-BA-57-E9 |
| 4 | /AMPUSER/100/HUSSEIN LEMA/MAC | E2:5C:49:7C:1A:4D |

Figure 3. Data stored in the astDB database.

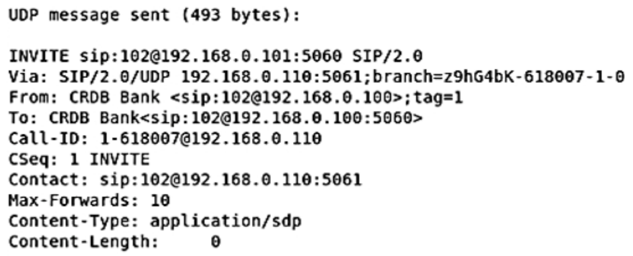
The metrics used to analyze the performance are the number of detected spoofed CID names, and the size of SIP request messages. Also, the SSD before and after the implementing algorithm is evaluated. In addition, a warning to the recipient without disclosing the privacy of the callers is captured. A warning is shown on the softphone screen as an alert/recovery to the callee.

## 6. Results

The result of the testbed experiment is a display name, which appears on the softphone screen regardless of whether the caller fakes the CID name or not. In addition, we analyze the size of packets after and before inserting the MAC address by using the SIPp log file. Finally, the SSD was calculated before and after using a developed algorithm.

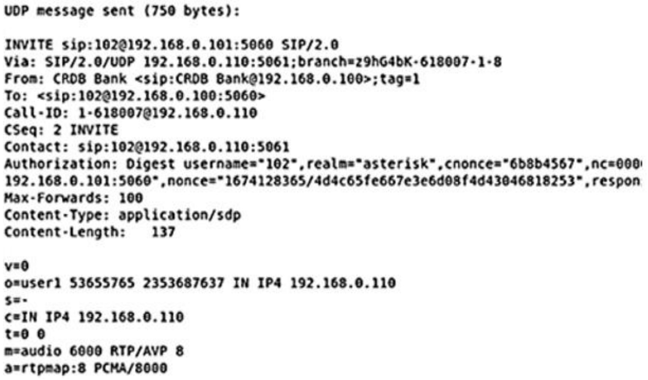
Appendix J shows the appearance of the CID name on the screen in the From field extracted from Wireshark on the SIP flow section. It shows CID names with a correct MAC address, CID number and CID name. The CID number must be validated because, if there is no display name, the default display name is the CID number. The display name was shown as it is if the CID name and MAC address are correct or the algorithm was not used to validate CID name as appeared in appendix K. Otherwise, the display name is displayed as it is. On this solution, all spoofed CID names are detected, and an alert is made to the callee.

In evaluating the size of packets, the study looked the size of INVITE packets before and after MAC addresses were included. Figure 4 shows the size of the INVITE request message is 493 bytes, while Figure 5 has 750 bytes because it has authorization information.



```
UDP message sent (493 bytes):

INVITE sip:102@192.168.0.101:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.110:5061;branch=z9hG4bK-618007-1-0
From: CRDB Bank <sip:102@192.168.0.100>;tag=1
To: CRDB Bank<sip:102@192.168.0.100:5060>
Call-ID: 1-618007@192.168.0.110
CSeq: 1 INVITE
Contact: sip:102@192.168.0.110:5061
Max-Forwards: 10
Content-Type: application/sdp
Content-Length:     0
```

Figure 4. Size of INVITE request message without MAC address and Authorization information.

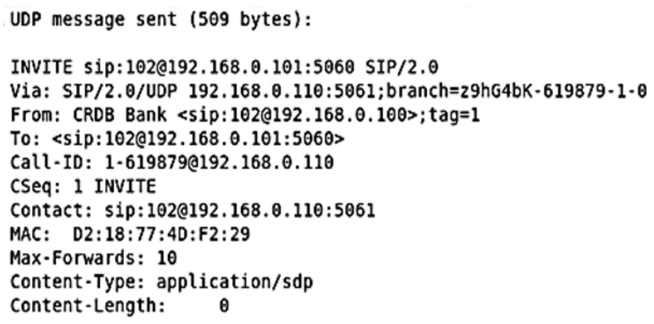


```
UDP message sent (750 bytes):

INVITE sip:102@192.168.0.101:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.110:5061;branch=z9hG4bK-618007-1-8
From: CRDB Bank <sip:CRDB Bank@192.168.0.100>;tag=1
To: <sip:102@192.168.0.100:5060>
Call-ID: 1-618007@192.168.0.110
CSeq: 2 INVITE
Contact: sip:102@192.168.0.110:5061
Authorization: Digest username="102",realm="asterisk",cnonce="6b8b4567",nc=000
192.168.0.101:5060",nonce="1674128365/4d4c65fe667e3e6d08f4d43046818253",respon
Max-Forwards: 100
Content-Type: application/sdp
Content-Length:   137

v=0
o=user1 53655765 2353687637 IN IP4 192.168.0.110
s=-
c=IN IP4 192.168.0.110
t=0 0
m=audio 6000 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

Figure 5. Size of INVITE request message without MAC address but with authorization information.

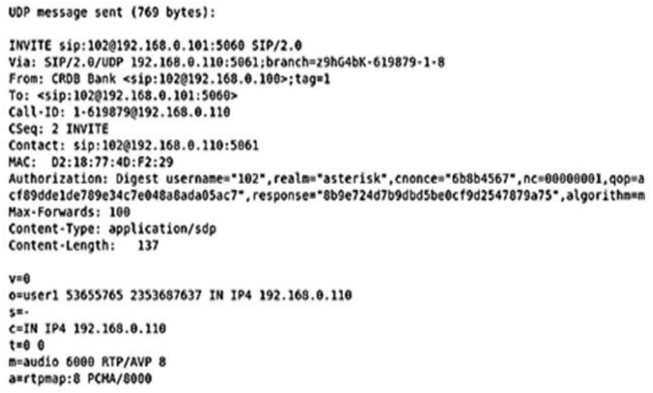
In addition, Figure 6 shows that the size of the INVITE request message is 509 bytes with a MAC address, while Figure 7 shows the size of 769 bytes with a MAC address and authentication information. The difference in the sizes of the INVITE request messages between Figures 4 and 6 is 16 bytes, and Figures 5 and 8, the difference is 19 bytes, which is smaller than previous solutions that involved a completely new call/message to validate CID.

Finally, by using Wireshark and Microsoft spreadsheets, we calculated SSD before and after using the algorithm. Table 1 shows the results captured from Wireshark that were used to calculate SSD versus the call rate per second. The SSD, before inserting the algorithm, ranged from 15.8 ms to 27.2 ms after sending 10 INVITE request SIP messages with different call rates by using the SIPp simulator. All the packets were captured with the as if has correct CID name before using algorithm.



```
UDP message sent (509 bytes):

INVITE sip:102@192.168.0.101:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.110:5061;branch=z9hG4bK-619879-1-0
From: CRDB Bank <sip:102@192.168.0.100>;tag=1
To: <sip:102@192.168.0.101:5060>
Call-ID: 1-619879@192.168.0.110
CSeq: 1 INVITE
Contact: sip:102@192.168.0.110:5061
MAC:  D2:18:77:4D:F2:29
Max-Forwards: 10
Content-Type: application/sdp
Content-Length:     0
```

Figure 6. Size of INVITE request message with MAC address but without Authorization information.



```
UDP message sent (769 bytes):

INVITE sip:102@192.168.0.101:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.110:5061;branch=z9hG4bK-619879-1-8
From: CRDB Bank <sip:102@192.168.0.100>;tag=1
To: <sip:102@192.168.0.101:5060>
Call-ID: 1-619879@192.168.0.110
CSeq: 2 INVITE
Contact: sip:102@192.168.0.110:5061
MAC:  D2:18:77:4D:F2:29
Authorization: Digest username="102",realm="asterisk",cnonce="6b8b4567",nc=00000001,qop=a
cf89dde1de789e34c7e048a8ada05ac7",response="8b9e724d7b9dbd5be0cf9d2547879a75",algorithm=m
Max-Forwards: 100
Content-Type: application/sdp
Content-Length:   137

v=0
o=user1 53655765 2353687637 IN IP4 192.168.0.110
s=-
c=IN IP4 192.168.0.110
t=0 0
m=audio 6000 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

Figure 7. Size of INVITE request message with MAC address and Authorization information.

Table 1. Session Set-up delay before using the algorithm.

| Time to send INVITE | BEFORE INSERT MAC ADDRESS | | | |
|---|---|---|---|---|
| | Time to receive 180 | SSD | CPS | SSD (ms) |
| 09:39:46.29653 | 09:39:46.31433 | 00:00:00:017799 | 10 | 17.799 |
| 09:39:58.90665 | 09:39:58.92306 | 00:00:00:016413 | 20 | 16.413 |
| 09:40:08.03841 | 09:40:08.05597 | 00:00:00:017551 | 30 | 17.551 |
| 09:40:17.25407 | 09:40:17.27360 | 00:00:00:019533 | 40 | 19.533 |
| 09:40:26.41429 | 09:40:26.43614 | 00:00:00:021850 | 50 | 21.85 |
| 09:40:35.64173 | 09:40:35.66017 | 00:00:00:018437 | 60 | 18.437 |
| 09:40:44.91292 | 09:40:44.92927 | 00:00:00:016357 | 70 | 16.357 |
| 09:40:55.55735 | 09:40:55.57318 | 00:00:00:015827 | 80 | 15.827 |
| 09:41:05.11139 | 09:41:05.12888 | 00:00:00:017486 | 90 | 17.486 |
| 09:41:15.86512 | 09:41:15.89252 | 00:00:00:027322 | 100 | 27.322 |

Table 2 shows the results collected from Wireshark inserted into the spreadsheet to calculate the SSD. Ten INVITE requests SIP messages at different call rates were sent to the PBX. Results were collected after inserting the algorithm into the controller for capturing IP and MAC addresses. Also, the algorithm for validating CID was implemented in dialplan within the freePBX. SSD ranges from 15.1 ms to 18.15 ms. Calls made have either valid CID or invalid CID.

Table 2. Session Set-up delay after using the algorithm.

| Time to send INVITE | AFTER INSERT MAC ADDRESS | | | |
|---|---|---|---|---|
| | Time to receive 180 | SSD | CPS | SSD(ms) |
| 11:24:38.51721 | 11:24:38.53536 | 00:00:00:018142 | 10 | 18.142 |
| 11:24:55.99393 | 11:24:56.01280 | 00:00:00:018250 | 20 | 18.250 |
| 11:25:10.80619 | 11:25:10.82434 | 00:00:00:018149 | 30 | 18.149 |
| 11:25:19.54987 | 11:25:19.56729 | 00:00:00:017424 | 40 | 17.424 |
| 11:25:24.22898 | 11:25:24.24702 | 00:00:00:018035 | 50 | 18.035 |
| 11:25:46.88591 | 11:25:46.90313 | 00:00:00:017223 | 60 | 17.223 |
| 11:25:54.39409 | 11:25:54.41149 | 00:00:00:017397 | 70 | 17.397 |
| 11:26:08.82534 | 11:26:08.84435 | 00:00:00:019007 | 80 | 19.007 |
| 11:26:18.53374 | 11:26:18.54884 | 00:00:00:015001 | 90 | 15.001 |
| 11:26:26.73000 | 11:26:26.74781 | 00:00:00:017812 | 100 | 17.812 |

Appendix K shows some of the SIP packets captured by using the TCPdump command on SIP flow within the Wireshark. TCPdump captured all packets passed through all ports on the laptop, including the virtual host created by Mininet (h1_eth0). There are two sections; first section shows fake CID without using an algorithm, which shows the CID name without detecting and provides a warning to a victim. While, the section shows detected fake CID name and fake or without MAC address of the INVITE request messages. If there is no MAC address or fake MAC address, display name shown on the Linphone screen starting with the word “fake_MAC_ADDR”. But if the MAC address is correct but the CID name is fake, the display name shown on the Linphone screen starting with the word “fake”.

Figure 8 shows the relationship between call rate (multiple of 10) versus session setup delay before and after applying the algorithm. As shown in the graph, there is a minor difference between SSD before and after using the algorithm. It was approximately between 1 ms to 3 ms, which was independent of the presence of the algorithm because their variation is randomly distributed on both graphs. Therefore, the result of implementing the algorithm has a negligible effect on SSD. This result signifies a significant improvement compared with the previous solutions. The last solution differs by 27.125ms between normal communication and verification communication [11]. This is because there is no forward and backward communication during the validation process. It used the same SIP messages that were used to establish a call in validating CID.

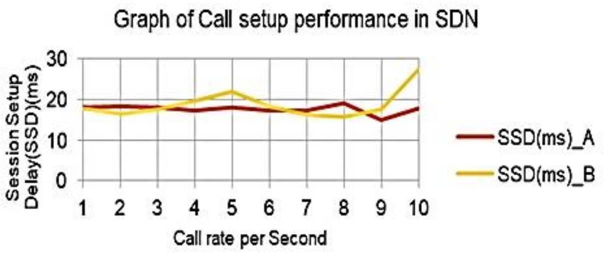


Figure 8. Graph of SSD before using the algorithm (SSD (ms)_B) and SSD after using the algorithm (SSD (ms)_A) Vs Call Rate per Second.

## 7. Discussions

As shown in the results, the performance measures are SSD, detection of CID spoofing, and warning provision to a callee by inserting the word fake. Also, the additional size of the packet to validate CID. All INVITE request SIP messages validate their CID name when run on the algorithm. The packets with a fake CID are detected and displayed as a fake CID name to warn/alert the callee/victim.

Session set-up delay ranges from 15.1 ms to 27.3 ms. These values are collected regardless of either the system has an algorithm or not. The average result of SSD is about 18 ms. This delay time is ranked as an excellent level of Quality of Service (QoS) for Internet communication [19] and it is in the acceptable range [31]. Therefore, the proposed solution produced minor differences of about 1 ms to 3 ms. Also, the result shows that the source of variation was not dependent on the validation process. Some related research reported packet loss, CPU utilization or effect description on delay. Also, they reported the usage of shared networks. For example, CallerDec [4, 18] "can cause the DoS and has an average verification delay was about 8.40 seconds." NASCENT [9] measured packet loss and CPU utilization. Also, it uses information from SIM cards like IMSI and MSISDN to validate CID. Also, a solution made by S. A. Iranmanesh provides an additional of delay about 27.125 ms [11] in validating the CID name.

The study used pure VoIP communication by using SDN technology with freePBX. Also, it used softphones within smartphones and laptops to conduct testbed experiments. Therefore, it used the internet with an asterisk server to run the algorithm without a SIM card. The study used the MAC address by adding it to the INVITE/REGISTER request message as a substitution for IMSI and MSISDN. The previous solutions used new calls or messages to validate CID [4, 9, 28]. These calls/messages analyze status or search caller details to the PBX and return to the callee. Others use information which is not available to all network devices, such as IMSI and MSISDN. Therefore, the proposed solution can be conducted with any devices that use the Internet. Previous solutions need specific devices or upgraded devices to support the solution.

With deployment, the study only used code/software to make validation within the existing infrastructure. It added a few sizes of storage within the controller on the LAN network. Therefore, its deployment is higher compared to the previous solutions. For example, CallerDec and CEIVE applications use an Android application to collect caller details with a specific setup for the provider to accept additional calls/messages [5]. But, iVisher and NASCENT need an upgrade of some network components, such as gateway, to communicate with each other. Also, they upgrade some components to make them programmable to enable CID validation [9].

Most solutions use fields that depending on specific devices or they are not unique to all devices. For example, the CallerDec application and CEIVE use call/message verification. Then, a caller should respond accordingly to give a callee his or her status/information. The NASCENT system uses IMSI and MSISDN, which are unique for a single SIM Card [9]. However, it is not used by all devices. Some devices can communicate on the internet without using a SIM Card. The study used the MAC address as a standard unique identification of all local network devices.

Some solutions are activated either by the callee (end-user) or by the gateway on the receiving side. On the receiving side, there is limited information to authenticate the CID name. The study made validation on the Caller side. By collecting and storing the IP and MAC addresses as unique identifiers on the caller side before starting a call. Then, take that information to the PBX during registration of the UAC. Finally, validation was conducted during the call setup by using the CID number, CID name, and MAC address. Also, the proposed solution can work with no need for new hardware or upgrade of existing devices [9], [22].

This study used an INVITE message by adding the MAC address of the network devices. It enabled to validation of CID by using the same procedure as the original call without adding new messages or calls to validate CID. Therefore, the proposed solution was not affected by the call setup delay because the validation was made before the media flow started. Also, validation is made by using the same message used to start the call. The previous solutions use new messages or calls to make a validation process.

The study used about 16 bytes to 19 bytes as an additional size of packets by adding a MAC address to an INVITE request message. Therefore, the proposed solution used fewer bandwidths than most of the previous solutions. This is because most previous solutions use new calls or messages. A new call or message involves a sequence of messages in which they can generate 1000 bytes or more. For example, the complete call may involve INVITE (716 bytes), 100 TRYING (260 bytes), 180 RINGING (260 bytes), 200 OK response (100 bytes), and ACK (60 bytes) messages. Therefore, all mechanisms that involve new calls use not less than 1000 bytes. Hence; they use more shared network resources.

## 8. Proposed Solution Contribution

The proposed solution enhanced the security of the SIP protocol by embedding the MAC address to the SIP header. It can be used for different purposes in VoIP communication, including the validation of CID. Other solutions use fields which are not unique, which cause the generation of new calls or messages or produce unexpected results [4, 18].

The proposed solution enabled the validation of CID names with low utilization of shared network resources. It needs an additional of about 24 bytes, while most previous solutions regenerate new calls or messages. Those new messages or calls increased the number of bytes within the network during the validation process [4, 18, 28, 38]. Hence, they use more shared network resources.

The proposed solution started validation at the initial stage (call setup) and location (in LAN on the caller side). But most of the existing solutions started validation either on the callee side [19], [5], [20] or within GW [9], [22].

The proposed solution developed an algorithm without the addition of any new devices or the upgrade of existing devices. With the use of SDN technology, the study enabled validation of the CID by using Layer 3 switches, POX controller [20] and freePBX. The added algorithm enabled the system to validate the CID name. Other solutions need new devices or upgrades to some of the existing devices to enable them to validate CID [26, 28]. Also, some solutions need a special setup for the telephony providers and involve both caller and callee to validate CID [4, 18, 38]. Appendix G shows how the proposed solution improved the solution with comparison to previous solutions.

## 9. Conclusion

Based on the study observations, CID spoofing is among the challenging problems because, in one way, it is legal and provides privacy to the users. On the other way, it can be used illegally and result in loss of money and confidential information. Also, defending against CID spoofing is not a simple task because VoIP uses the internet. Also, SIP messages do not have enough information about a caller to enable validation of CID anywhere.

The proposed solution is a perfect solution that protects users against illegal usage of CID spoofing. Showing the display name on the screen on either a real or fake name will alert a callee about their communication. Therefore, it gives the caller the privacy to use any name. But it will alert the callee that the name he/she is communicating with is a real or fake name. Therefore, if the CID name is fake, the callee cannot trust a caller because of his or her forgery of the CID name. The proposed solution can be integrated with an encrypted mechanism to protect SIP contents from external hackers. This work could not work on encryption mechanism because it works within the LAN.

The proposed solution produced better results than the previous result. It minimizes the usage of shared resources (bandwidths) by using only 24 bytes extra to validate CID. The proposed solution used existing infrastructure with no need for an upgrade. However, most of the previous solutions produce new calls or messages to validate CID, and use 1000 bytes or more. Therefore, they use more shared network resources.

Also, the proposed solution produces a better session set-up delay than most of the previous solutions. It ranges from 15.1 ms to 27.3 ms, which minimises the call setup delay of an original call [32]. The previous solutions withhold or leave an original call to continue ringing and start a new call or message. Then, they wait for the response of the new call/message made by the callee or GW/IMS. The most efficient previous solution added 27.135 ms as compared to normal communication. The proposed solution added 1 ms to 3ms as compared to normal communication.

We recommend that, for the CID solution to work better, every user/organization should protect themselves from spoofing their CID name. To work well, the providers of the services (VoIP services) should validate their customers by using the proposed solution on dialplan. It can assist the provider in knowing exactly where the problem started. Also, it can help in searching for an individual who is making spoofing through the MAC address.

If there is a hacker in the middle, it could be difficult for the solution to detect. The proposed solution recommends the usage of other mechanisms to detect the Man in the Middle attacks. Service providers can use alternative methods to detect man-in-the-middle attacks on mobile data devices. Mobile data users can use the same study solution by using another way to extract the MAC address [24]. But, the validation of the CID name on the dialplan can use the same procedure as proposed in the study for the service provider.

Also, every part of the network operator should be responsible for its part. This includes organizations/institutions, network/service providers, and the government. For example, every organization should validate their caller before sending it to the receivers and insert the MAC address into the SIP header. Then, the providers can use the SIP header information to validate the CID name. Finally, the government should make a policy/law. These laws/policies will require organizations and service providers to monitor their customers.

Therefore, the SIP protocol was enhanced by adding a MAC address, which is a unique identity of the devices used to communicate on the internet (LAN network). It validates/identifies hackers within the SIP server or in any location where the system can read the SIP header. In our case, we used it to validate CID names and numbers and produce better results than previous solutions. It reduced intercommunication to search for caller information and provide a warning to the callee by using the word “fake”.

The proposed solution lack on defending against man in the middle attack. Therefore, it was suggested to use any encrypted algorithm to protect SIP contents of being changed; even though, originally it was made to transmit data in plain text. Another way was to encrypt some potential data by hash function like MAC address, CID name and any other important information that hackers can use them to risk victims. This inquiry will be our future work.

## ACKNOWLEDGEMENT

We thank the University of Dar es Salaam for financial support and all reviewers and editor of this journal for their good comments, which shaped our work.

## CONTRIBUTIONS OF CO-AUTHORS

| | | |
|---|---|---|
| Hussein S. Lema | [ORCID: 0000-0003-4796-4723] | Conceived the idea, conducted experiments, analyzed results and wrote the paper. |
| Dr. Fatuma Simba | [ORCID: 0000-0002-6574-759X] | Provided technical information on methods and materials, reviewed the paper for correction, comments, and inputs. |
| Dr. Joseph C. Mushi | [ORCID: 0000-0003-4095-0433] | Provided technical information on methods and materials, reviewed the paper for correction, comments, and inputs. |

## APPENDICES

### Appendix A: Summary of the steps used in Design Science Research Methodology

Table A.1. Summary of the steps used to develop the proposed solution as defined by DSRM

| Research Step | Concerns | Output to Next Step | Entry Point? |
|---|---|---|---|
| Identify the Problem & Motivate | SIP protocol has a little information to validate call ID and its messages are editable. Hacker can change CID to falsify the callee.<br>VoIP is the cheapest in audio/video calls, with attractive features. | Create an algorithm to validate CID that can minimize usage of shared network resources, call setup delay without degrading quality of VoIP services. | Change CID name, made a callee trust a caller. Hence, the callee can expose his/her valuable material/ information. |
| Define Objectives of a Solution | Validate CID through SDN by using the MAC address to minimize loss of valuable material. | Existing solutions use more shared network resources with high call setup delays. To work well, need upgrade of some tools. Hence, the proposed solution mitigates those flaws. | To enhance SIP protocol by adding MAC address to validate CID to mitigate CID spoofing attack. |
| Design & Development | Design and develop algorithms that can validate CID with minimum delay and usage shared resources. | To run an algorithm by making an experiment on virtual and real devices to analyze RRD, SRD and detection of fake CID names. | By using a designed topology, the experiment evaluates the performance of the algorithm |
| Demonstration | Experiment is conducted by using Mininet, POX controller, SIPp simulator, and freePBX. It is tested with and without algorithms and compares their result. | VoIP performance was analyzed. Packets captured on Wireshark, log files of the SIPp simulator and POX controller are used to collect data. | The study used RRD, SRD and detection of CID name on the Linphone screen to analyze the algorithm. |
| Evaluation | Detected fake CID shown on the screen with the word fake. The RRD is about 10.1ms and the SRD ranges between 15.1ms to 27.3 ms. | The performance of VoIP is in the acceptable delay range under the minimum usage of a shared resource during the validation process. | The study decreased delay, hence improving VoIP services. |
| Communication | Published to Journal of ICT systems (jicts) and presented to the departmental meeting. | Gain comments and different help to improve our paper. | Improved results and thesis report. |

### Appendix B: Code segment to create virtual topology in a Mininet emulator

```
info( ‘*** Add hosts\n’)
.
   net.addLink(h1, s1)
   net.addLink(h2, s1)
 info( ‘*** Starting network\n’)
   net.start()
   h1.cmdPrint(’dhclient ‘+h1.defaultIntf().name)
   h2.cmdPrint(’dhclient’+h2.defaultIntf().name)
```

## Appendix C: Code segment to collect IP and MAC address to the CSV file

```
  if isinstance(packet.next, ipv4):
    path = ‘/home/hussein/sipp/script/REGISTER_client_caller.csv’
    file = pathlib.Path(”/home/hussein/sipp/script/REGISTER_client_caller.csv”)
    if file.exists():
       log.info(”file exists “)
       if os.path.getsize(’/home/hussein/sipp/script/REGISTER_client_caller.csv’) == 0:
         log.info(”file is ampty “)
         with open(’/home/hussein/sipp/script/REGISTER_client_caller.csv’, ‘a’) as file:
           writer = csv.writer(file)
           writer.writerow([packet.next.srcip, packet.src])
           log.info(”Record is recorded to a file with IP address and MAC address “)
       else:
          with open(’/home/hussein/sipp/script/REGISTER_client_caller.csv’, ‘r’) as csvfile:
              spamreader = csv.reader(csvfile, delimiter = ‘;’, quotechar = ’, quoting =
csv.QUOTE_MINIMAL)
                  …..
                   writer = csv.writer(file)
                   writer.writerow([packet.next.srcip, packet.src])
                   log.info(”Record is recorded to a file with IP address %s and MAC address is %s “,
packet.next.srcip, packet.src)
    else:
      log.info (”File not exist”)
      with open(’/home/hussein/sipp/script/REGISTER_client_caller.csv’, ‘a’) as r:
         log.info(”file is created “)
```

## Appendix D: Code to create Register/Invite requests messages

```
INVITE sip:[field0]@[remote_ip]:[remote_port] SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
From: [field3] <sip:[field0]@[field1]>;tag=[call_number]
To: [field3]<sip:[field0]@[field1]:[remote_port]>
Call-ID: [call_id]
CSeq: [cseq] INVITE
MAC:[field4]
Contact: sip:[field0]@[local_ip]:[local_port]
Max-Forwards: 10
```

## Appendix E: Code that validate CID name in Dialplan

```
[from-internal-custom]
exten => _1XX,1,Set(from_header=${PJSIP_HEADER(read,From)})
exten => _1XX,n,Set(MAC_header=${PJSIP_HEADER(read,MAC)})
exten => _1XX,n,GotoIf($[${DB_EXISTS(AMPUSER/${CALLERID(num)}/
${CALLERID(name)}/MAC)} != 1]?add_to_DB:validate_all)
exten => _1XX,n(add_to_DB),Set(DB(AMPUSER/${CALLERID(num)}/
${CALLERID(name)}/MAC)=${MAC_header})
```

```
exten => _1XX,n,Set(MAC_address=${DB(AMPUSER/${CALLERID(num)}/
${CALLERID(name)}/MAC)})
exten =>  _1XX,n,Goto(validate_all)
exten => _1XX,n(validate_all),GotoIf($[”${DB(AMPUSER/${CALLERID(num)}
/${CALLERID(name)}/MAC)}” = “${MAC_header}”]?mac_is_ok:mac_is_not_ok)
exten => _1XX,n(mac_is_not_ok),Set(CALLERID(name)=Fake_MAC_ADDR- ${display_name})
exten => _1XX,n(mac_is_ok),GotoIf($[${display_name}=”${CALLERID(name)}” &
${caller_number}=${CALLERID(num)]?Unspoofed_call:spoofed_call)
exten => _1XX,n(spoofed_call),Set(CALLERID(name)=Fake-${display_name})
exten => _1XX,n,SendText(Umefake caller id mane)
exten => _1XX,n,Dial(${PJSIP_DIAL_CONTACTS(${EXTEN})})
exten => _1XX,n(Unspoofed_call),Dial(${PJSIP_DIAL_CONTACTS (${EXTEN})})
```

## Appendix F: Data from CSV storage saved to a POX controller

```
SEQUENTIAL
100;192.168.1.108;[authentication username=100 password=100];HUSSEIN;192.168.1.110;
D2:18:77:4D:F2:29;
103;192.168.1.108;[authentication username=103 password=103];OMARY;
192.168.1.100;D8:D0:90:36:4E:7C;
```

## Appendix G: Performance of the solutions on CID spoofing attacks

Table G.3. Evaluation of proposed solution with previous solutions

| SOLUTIONS | Parameter used to analyse performance of the solution on CID spoofing | | | |
|---|---|---|---|---|
| | Packet Size | SSD | Detection of CID spoofing | DoS attacks |
| Proposed solution | Added 24 Byte | 20milisecond(ms) | Used word Fake to warn & detect all fake CID | No DoS attack. |
| CIV [20] | Need more than 1MB | 20 seconds(s) | Use caller response to detect fake CID. Not detect all fake CID. | Can be used to generate DoS attacks |
| CallerDec [5] | Need more than 1MB | 8.40 s | Use caller response to detect fake CID. Not detect all fake CID. | Can be used to generate DoS attacks |
| CEIVE [19] | Need more than 1MB | 4-10 s | Use caller response to detect fake CID. Not detect all fake CID. | Can be used to generate DoS attacks |
| STIR/SHAKEN[25] | Need more than 1MB | Not analyzed | Detect only the Caller carrier. | Not analyzed |
| iVisher [22] | Need more than 1MB | 994 ms | Provide caller detail in text/audio to the callee. | Can be used to generate DoS attacks |
| NASCENT [9] | Need more than 1MB | 18 ms | Detect all fake CID with 12% to 19% call drop | No DoS attack |

## Appendix H: Algorithm that was used to validate CID name.

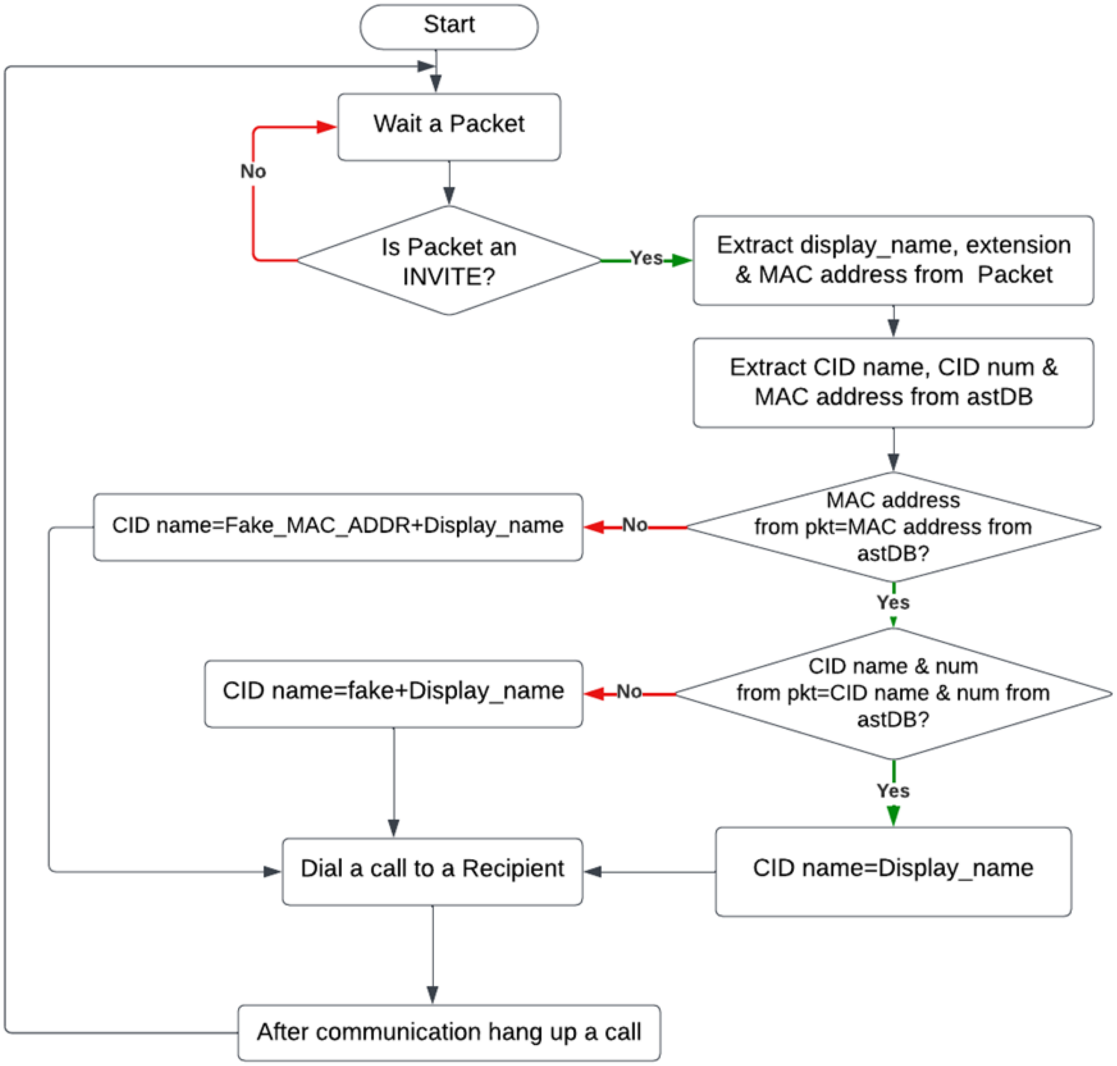

Start
Wait a Packet
No
Is Packet an INVITE?
Yes
Extract display_name, extension & MAC address from Packet
Extract CID name, CID num & MAC address from astDB
MAC address from pkt=MAC address from astDB?
No
CID name=Fake_MAC_ADDR+Display_name
Yes
CID name & num from pkt=CID name & num from astDB?
No
CID name=fake+Display_name
Yes
CID name=Display_name
Dial a call to a Recipient
After communication hang up a call

Figure H.1. Algorithm to validate CID name within the dialplan

## Appendix I: Algorithm that collects and store IP and MAC addresses

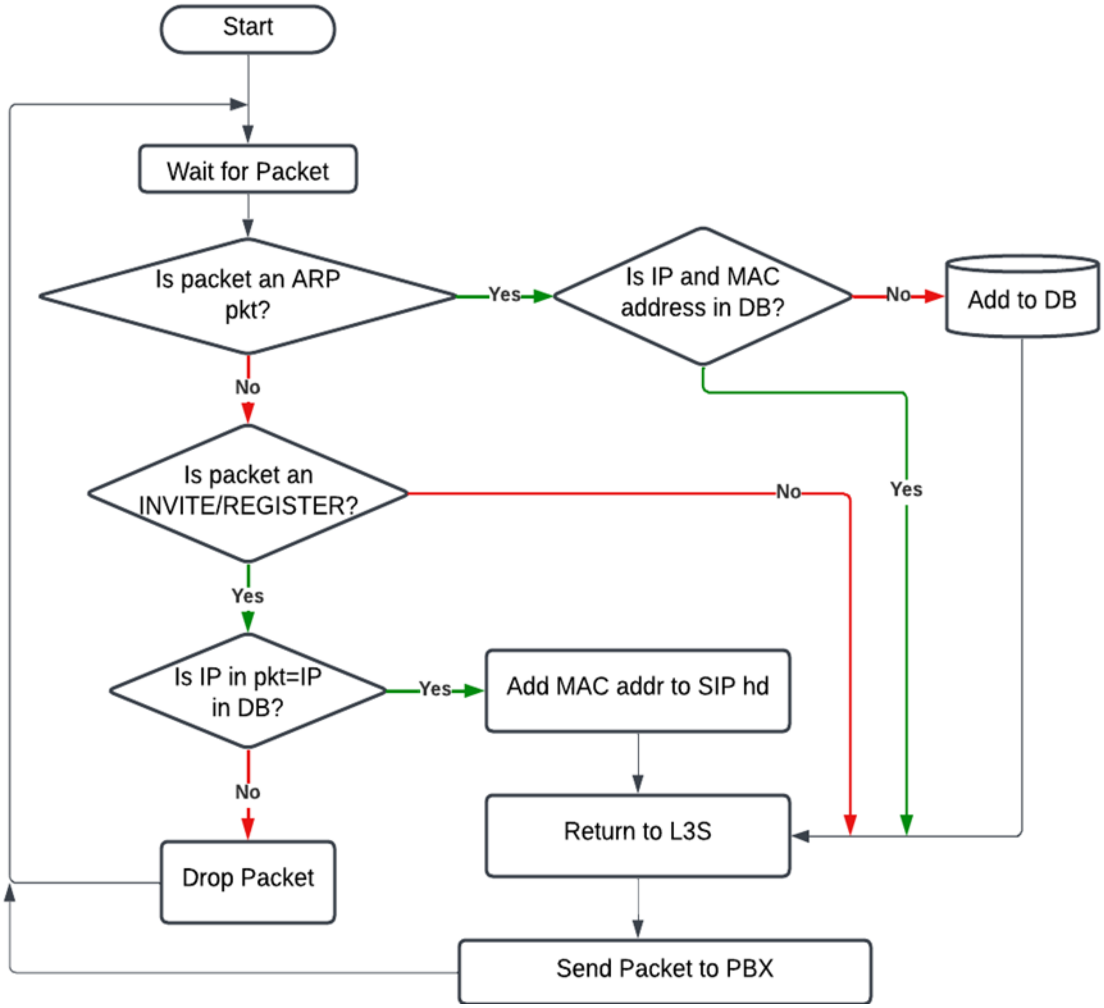


Figure I.1. Algorithm to collect IP and MAC address to the POX controller

## Appendix J: Result from SIP flow with correct CID name

Table J.1. Data collected from SIP flow with correct CID name

| RESULT OF ALGORITHM WITH CORRECT CID NAME | | | | | | |
|---|---|---|---|---|---|---|
| Start time | Stop Time | Initial speaker | From | To | State | Comments |
| 9.073850 | 11.422823 | 192.168.0.104 | HADIJA LEMA <sip:102@192.168.0.100> | <sip:102@192.168.0.101:5060> | COMPLETED | INVITE 401 200 |
| 11.473958 | 13.814123 | 192.168.0.104 | MUSTAFA RAJABU“ <sip:104@192.168.0.104> | <sip:104@192.168.0.101:5060> | COMPLETED | INVITE 401 200 |
| 11.473958 | 13.814123 | 192.168.0.104 | MUSTAFA RAJABU <sip:104@192.168.0.104> | <sip:104@192.168.0.101:5060> | COMPLETED | INVITE 200 |
| 24.673727 | 27.372255 | 192.168.0.104 | HADIJA LEMA <sip:102@192.168.0.100> | <sip:102@192.168.0.101:5060> | COMPLETED | INVITE 401 200 |
| 17.077975 | 17.106771 | 192.168.0.104 | HASSAN LEMA <sip:101@192.168.0.105> | <sip:101@192.168.0.101:5060> | CALL SETUP | OPTIONS 200 |

## Appendix K: Result from SIP flow with incorrect/fake CID name

Table K.1. Data collect from SIP flow with fake CID name before and after using algorithm

| RESULTS BEFORE USING ALGORITHM WITH FAKE CID NAME | | | | | | |
|---|---|---|---|---|---|---|
| Start time | Stop Time | Initial speaker | From | To | State | Comments |
| 31.029288 | 33.450951 | 192.168.0.118 | CRDB Bank <sip:102@192.168.0.100> | <sip:102@192.168.0.101:5060> | COMPLETED | INVITE 401 200 |
| 33.528903 | 35.926737 | 192.168.0.118 | CRDB Bank <sip:102@192.168.0.100> | <sip:102@192.168.0.101:5060> | COMPLETED | INVITE 200 |
| 36.028203 | 38.545186 | 192.168.0.118 | Customer Care <sip:101@192.168.0.105> | <sip:104@192.168.0.101:5060> | COMPLETED | INVITE 401 200 |
| 38.728694 | 41.334842 | 192.168.0.118 | Customer Care <sip:101@192.168.0.105> | <sip:104@192.168.0.101:5060> | COMPLETED | INVITE 200 |
| 41.428492 | 43.662486 | 192.168.0.118 | OMARY <sip:104@192.168.0.104> | <sip:101@192.168.0.101:5060> | CALL SETUP | OPTIONS 200 |
| 43.828564 | 46.103307 | 192.168.0.118 | OMARY <sip:104@192.168.0.104> | <sip:101@192.168.0.101:5060> | COMPLETED | INVITE 200 |
| RESULT AFTER USING ALGORITHM WITH INCORRECT/FAKE CID NAME AND MAC ADDRESS | | | | | | |
| 15.863528 | 19.607698 | 192.168.0.101 | "Fake_MAC_ADDR-\"HASSAN LEMA\"" <sip:101@192.168.0.101> | <sip:104@192.168.0.104> | COMPLETED | INVITE 200 |
| 113.490227 | 126.815702 | 192.168.0.101 | "Fake_MAC_ADDR-\"HUSSEIN LEMA\"" <sip:100@192.168.0.101> | <sip:104@192.168.0.104> | COMPLETED | INVITE 200 |
| 18.349219 | 20.729717 | 192.168.0.101 | "Fake-\"OMARY\"" <sip:104@192.168.0.101> | <sip:104@192.168.0.104> | COMPLETED | INVITE 200 |
| 23.545591 | 25.622080 | 192.168.0.101 | "Fake-\"HASSAN\"" <sip:104@192.168.0.105> | <sip:104@192.168.0.104> | COMPLETED | INVITE 200 |

## REFERENCES

[1] Suthar, D. and Rughani, P., *A Comprehensive Study of VoIP Security,* 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India: p. 812–817, 2020.

[2] Shaw, U. and Sharma, B., *A Survey Paper on Voice over Internet Protocol (VOIP), Int. J. Comput. Appl.*, **139**(2): p. 16–22, 2016.

[3] Hidayat, A. and Saputra, I., "*Implementation Voice Over Internet Protocol (VoIP) as a Communication Media between Unit at University Muhammadiyah Metro*, Int. J. Inf. Syst. Comput. Sci., **2**(2): p. 59-66, 2019.

[4] Niruj, K., *Studying the Architecture and Signaling Flow of SIP,* Bachelor's Thesis, Metropolia University of Applied Sciences, Metropolia. Accessed: Jul. 24, 2023. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/496236/Studying%20SIP.pdf?sequence=2

[5] Mustafa, H., Xu, W., Sadeghi, A., and Schulz, S., *End-to-end detection of caller ID spoofing attacks,* IEEE Trans. Dependable Secure Comput., **15**(3): p. 423–436, 2018.

[6] Li, J., Faria, F., Chen, J., and Liang, D., *A mechanism to authenticate caller ID,* Recent Advances in Information Systems and Technologies, **570**: p. 745–753, 2017.

[7] Jan, S., Qayum, F., and Khan, A., *SIP Issues and Challenges – A Scalable Three Factor Authentication Scheme*, Mehran Univ. Res. J. Eng. Technol., **39**(2): p. 287–309, 2020.

[8] Buriachok, V., Sokolov, V., and Dini, M., *Research of Caller Id Spoofing Launch, Detection, and Defense*, Cybersecurity: Education, Science, Technique, **3**(7): p. 6-16, 2020.

[9] Sheoran, A., Fahmy, S., Peng, C., and Modi, N., "*NASCENT: Tackling Caller-ID Spoofing in 4G Networks via Efficient Network-Assisted Validation*," *Proc. - IEEE INFOCOM*, p. 676–684, 2019.

[10] Martin, J., Rye, E., and Beverly, R., *Decomposition of MAC address structure for granular device inference*, in Proceedings of the 32nd Annual Conference on Computer Security Applications, Los Angeles California USA, p. 78–88, 2016.

[11] Iranmanesh, S., "Security Enhancements in Voice Over IP Networks." 2017. [Online]. Available: http://dx.doi.org/10.21220/s2-9m47-ej95

[12] Lu, Q., Qu, H., Zhuang, Y., Lin, Y., and Ouyang, Y., *Client-Side Evil Twin Attacks Detection Using Statistical Characteristics of 802.11 Data Frames,* IEICE Trans. Inf. Syst., E101.D(10): p. 2465-2473, 2018.

[13] Alotaibi B., and Elleithy, K., *A New MAC Address Spoofing Detection Technique Based on Random Forests,* Sensors, **16**(3), p. 281, 2016.

[14] Rawat, D., and Reddy, S., *Software Defined Networking Architecture, Security and Energy Efficiency*: *A Survey*, IEEE Commun. Surv. Tutor., **19**(1): p. 325-346, 2017.

[15] Haji, S., *et al.*, *Comparison of Software Defined Networking with Traditional Networking,* Asian J. Res. Comput. Sci.Info. Tech., 9(2): p. 1–18, 2021.

[16] Isyaku, Mohd Zahid, B., Bte Kamat, M., Abu Bakar, K., and Ghaleb, F., *Software Defined Networking Flow Table Management of OpenFlow Switches Performance and Security Challenges: A Survey*, Future Internet, **12(**9): p. 147, 2020.

[17] Cabaj, K., Wytrębowicz, J., Kukliński, S., Radziszewski, P., and Dinh, K., *SDN Architecture Impact on Network Security*, Federated Conference on Computer Science and Information Systems, **3**: p. 143–148, 2014.

[18] Lema, H., Simba, F., and Ally, A., *Preventing Utilization of Shared Network Resources by Detecting IP Spoofing Attacks through Validation of source IP Address*, IST-Afr. Week Conf., p. 1–8, 2018.

[19] Deng, H., Wang, W., and Peng, C., *CEIVE: Combating Caller ID Spoofing on 4G Mobile Phones Via Callee-Only Inference and Verification*, in Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi India, p. 369–384, 2018.

[20] Wang, S., Delavar, M., and Azad, M., *Spoofing Against Spoofing: Towards Caller ID Verification In Heterogeneous Telecommunication Systems,* ACM Transactions on Privacy and Security, 27(1): p. 1-25, 2023.
[21] Mustafa, H., Xu, W., Sadeghi, A., and Schulz, S., *You Can Call but You Can't Hide: Detecting Caller ID Spoofing Attacks*, 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Atlanta, GA, USA, p. 168-179, 2014.
[22] Song, J., Kim, H., and Gkelias, A., *iVisher: Real-time detection of caller ID spoofing*, ETRIJ., **36**(5): p. 865–875, 2014.
[23] Mahsa,H., Yaghmaee, M., Seno, S., Roshkhari, H., and Asadi, M., *Anomaly-based DoS detection and prevention in SIP networks by modeling SIP normal traffic*" Int. J. Comm. Sys., **31**(18): p. 1-26, 2018.
[24] Ichsan, A., and Riadi, I., *Mobile Forensic on Android-based IMO Messenger Services using Digital Forensic Research Workshop (DFRWS) Method*, Int. J. Comput. Appl., **174**(18): p. 34–40, 2021.
[25] Yu, J., "*An Analysis of Applying STIR/SHAKEN to Prevent Robocalls,* Advances in Security, Networks, and Internet of Things, p. 277–290, 2021.
[26] Sulafa, T., and Barry, B., *Evaluating the Impact of Encryption on Voice over Internet Protocol (VoIP) Systems*, 2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE), Khartoum, Sudan, p. 686-691, 2013.
[27] Shen, C., Nahum, E., Schulzrinne, H., and C. Wright, C., *The impact of TLS on SIP server performance,* Proc. IPTComm 2010 - Princ. Syst. Appl. IP Telecommun, p. 59–70, 2010.
[28] National Cyber Security Center, *IT Security Guidelines for Transport Layer Security (TLS)*, The Hague, 2020.
[29] Venable, J., Pries-Heje, J., and Baskerville, R., *Choosing a Design Science Research Methodology*, ACIS2017, p. 1-11, 2017.
[30] Nosrati, M., *Python: An appropriate language for real world programming*, World applied Programming, **1**(2): p. 110–117, 2011.
[31] Samrah, A., Khalil, A., and Ibrahim, H., *Improving quality of service for internet protocol television and voice over internet protocol over long-term evolution networks,* Int J Commun. Netw. Distrib. Syst., **22**(4): p. 409–445, 2019.
[32] Singh, B., and Hans, R., *TCP and UDP Based Performance Analysis of AODV, DSR and DSDV Routing Protocols under Different Traffic Conditions in Mobile AdHoc Networks,* Int. J. Future Gener. Commun. Netw., **8**(2): p. 73–92, 2015.