# Threat Modelling & Security Analysis for Robot and Service

*Shabnaz Khanam*

*Masters in Communication Systems and Network*
*Cologne University of Applied Sciences*
*Betzdorfer Str. 2, 50679 Köln ,Germany*

*Shabnaz.khanam@smail.th-koeln.de*

**Abstract – Through the attack, the security level and vulnerabilities of new technology can be highlighted. Malicious and other attacks can create a significant damage of the system. Due to implications of threat on the system, products lose their authenticity. Now it is a popular topic to research. At present the risk of such attacks are exacerbated in modern technology such as Robot. In this project we will discuss about the new protocol Neuropil and its implementation. To define its security architecture, a threat modelling is used. In first half of this report, the communication structure and its security objects are explained and last half is about the application of one popular attack and its overcome techniques.**

**Keyword-Threat modelling, Robot, Neuropil, Authentication, Verification, DoS attack, Bloom filter.**

## I. Introduction

A robot is an automated machine that can perform specific tasks with speed and precision and with minimal or no human intervention[1]. The term robot can refer to a physical robot as well as a virtual software agent. Software robots have a huge market potential and are expected to grow every day. While the robotic automation of processes has significantly increased a company's productivity, the system has additional security vulnerabilities that allow attackers to easily gain access to all information and control. For the development of the security level against threats, the documentation of the analysis is necessary. The model created for analysing the various business and technical requirements of a system, which is identifying potential threats and documenting them is known as a threat model. The purpose of the threat modelling is to get a clear picture of the resources and the potential threats to the resources, and to figure out how and when to mitigate those threats. The result of a threat modelling is a robust security system.

In this paper, we will focus on security architecture by implementing attack. The paper is structured as follows. Section 2 focus on the Object , on section 3 is about System Processing, section 4 is about Identify threats, section 5 about Risk Mitigation Strategies, section 6 about Attack, and the last section about discussion and future work.

## II. Define The Object

The aim of this project is to define the security mechanism of the system where the implemented protocol claims that the stable communication is established between application and system with balance their privacy .To define its security an attack is applied.

## III. System Processing

### Define Data Flow Diagram (DFD)

Nowadays Industrial Robots are more intelligent compared to the past. Update configuration and machine learning techniques makes Robots faster and more skilled. It can calculate its own movements between two positions for that reason it is able to capture the sudden change in the field(example: sensor). It helps to avoid unexpected obstacles which is profitable and also harmful if it doesn't react in critical situations due to its vulnerabilities[25]. In this project we proposed about the control of the robot which depends on two other identities named Co-ordinator and Client.

In order to access the robot, the client needs permission and it sends the request to the co-ordinator. Co-ordinator verifies the authentication of the client ID, accepts the request and forwards it to the robot. Before granting permission, the coordinator sends the robot a claim message to introduce itself as the authority . To react to a client's action it depends on Co-ordinator response. So The robot receives this permission message from the coordinator, specifying the client ID and the body groups that the client is allowed to access. The robot responds and moves its allowed body parts according to the client's action message. Without permission, the robot denies the client's action request. All messages are transmitted after exchanging their status and the robot informs other identities of the network about its current situation through an update message.

[1] https://www.techtarget.com/searchenterpriseai/definition/robot

The nodes are used for establishing connections between hosts and the network. The nodes are known to their neighbouring nodes and send messages using the User Datagram Protocol.

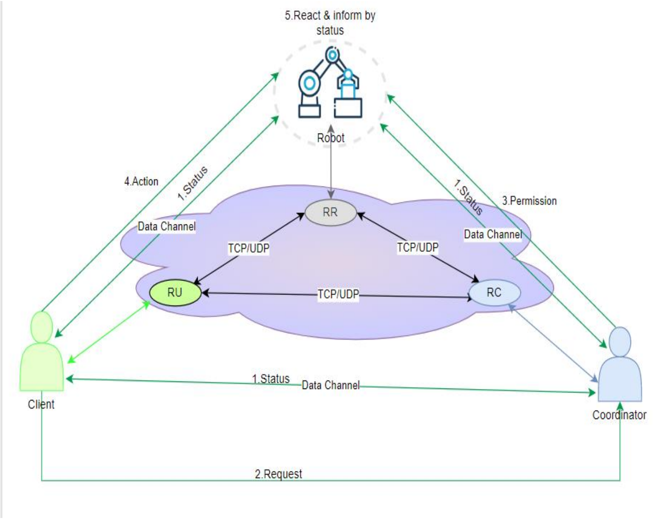


Fig1 : System Overview

## IV. Identify Threats

After reviewing the system, the question is marked on the security object to define data confidentiality and accuracy during transmission. Attackers can gain information or reuse/modify the data if there are insufficient security mechanisms. For low mechanisms data loses its security objects. The table explains what types of threats data is exposed to.

| Data | Lacking of Security Object | Threat |
|---|---|---|
| Status | Integrity | Increased Maintenance cost(From business perspective) |
| | Authentication | Losing identification (anyone can connect ) |
| Request Message | Integrity | Third party Access |
| | Authorization | Denial of Service attack |
| Permission Message | Authentication | Losing identification |
| Action Message | Authorization | Denial of Service attack |
| | integrity | Increased Maintenance cost |
| | freshness | Replay Attack |
| | Non-repudiation | Repudiation(attack deny the action) |

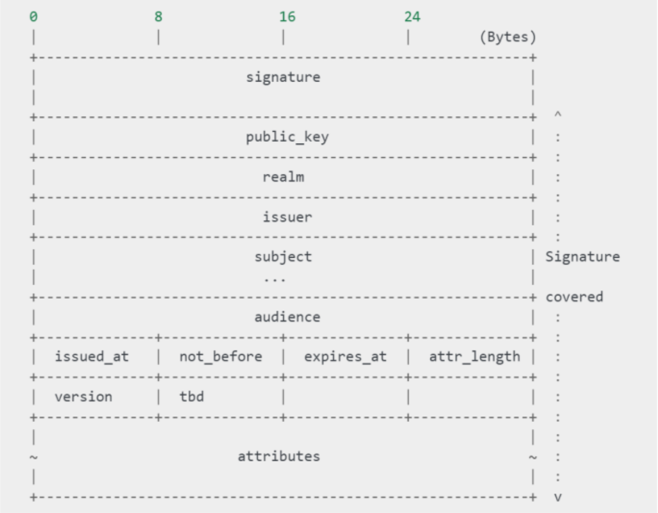
Table1: Message with threat[1]

## V. Risk Mitigation Strategies

As a process, risk mitigation begins with the generation of attack trees [3]. Once the strategy and protocol of the system is explained, then the threat will be identified more clearly. The more detailed the data flow diagram and thus the individual components of the protocol are, the better mitigation strategies can be developed.

### 5.1 Protocol

The system is using Neuropil protocol which is new protocol and structured with a combination of the concept of data network, self-sovereign identities, zero trust architectures and attributes-based access control [2]. Following self-sovereign [15,16] identities Neuropil use token considering as an identity. There are three kinds of token[2], and every token is used with specific reason. Establishing the communication one token use other tokens such as the message intent token which is issued by an identity token and refers to the node token. For the information it contains, a token is unique and different from others, and it cannot be changed unless the owner wants it to be. The information of token is given in fig 2.



```
0               8               16              24
|               |               |               |        (Bytes)
+---------------------------------------------------------------+
|                           signature                           |
|                                                               |
+---------------------------------------------------------------+  ^
|                          public_key                           |  :
+---------------------------------------------------------------+  :
|                             realm                             |  :
+---------------------------------------------------------------+  :
|                            issuer                             |  :
+---------------------------------------------------------------+  :
|                            subject                            | Signature
|                              ...                              |
+---------------------------------------------------------------+ covered
|                           audience                            |  :
+---------------+---------------+---------------+---------------+  :
|  issued_at    |  not_before   |  expires_at   |  attr_length  |  :
+---------------+---------------+---------------+---------------+  :
|  version      |  tbd          |               |               |  :
+---------------+---------------+---------------+---------------+  :
|                                                               |  :
~                          attributes                           ~  :
|                                                               |  :
+---------------------------------------------------------------+  v
```

Fig 2: The Token Structure[3]

[2] https://www.rfc-editor.org/rfc/rfc8392

(**UUID**) : Universal unique identifiers help to identify the entity with unique ID so that it is not possible to duplicate another one that already exists.

**Char Subject** defines what is this token about

**Char Issuer** indicates who issued this token

**Char Realm** is about who is the owner

**Char audience** is for the intended audience

**Double issued_at** means token creation time

**Double not_before** define when to be used

**Double expires_at** means validation time

**Unsigned char public_key** is public key of this token

**Unsigned char signature** denote signature of the token

### 5.1.1 Handshake Message

The main concept of Zero trust architecture [19,20] is "never trust, always verify". So before trusting other node, verification is important for communication. For the verification, Handshake token is exchanged with the core information of node to establish transport layer encryption. In general a Handshake message performs with three steps (SYN,SYN-ACK,ACK) but Neuropil protocol authentication is completed by four steps (exchanging handshake and join token). The first message which is sent is a Handshake token in plain text that contains the initiating node's fingerprint and signature, public key.

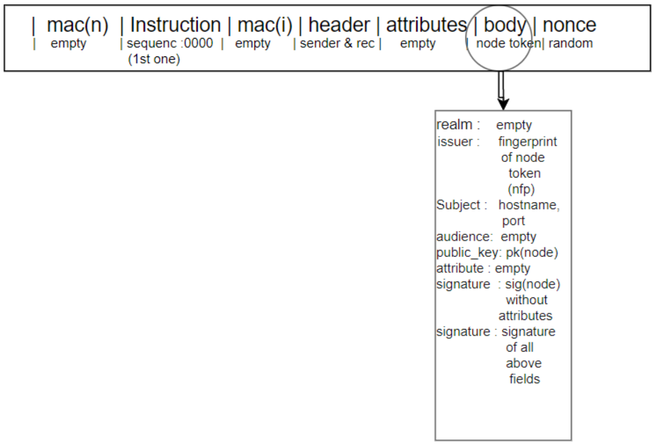

| mac(n) | Instruction | mac(i) | header | attributes | body | nonce
| empty | sequenc :0000 | empty | sender & rec | empty | node token | random
(1st one)
realm : empty
issuer : fingerprint of node token (nfp)
Subject : hostname, port
audience: empty
public_key: pk(node)
attribute : empty
signature : sig(node) without attributes
signature : signature of all above fields

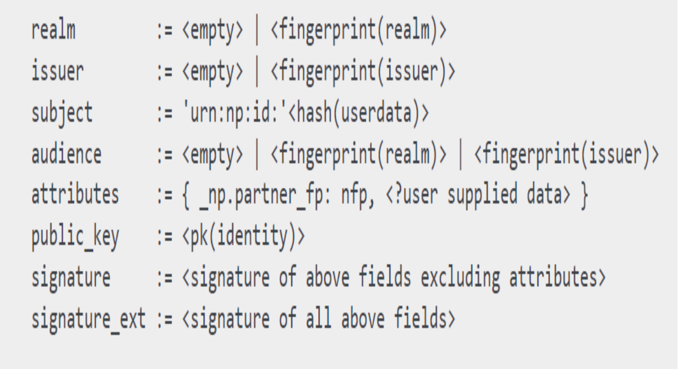
Fig 3:Handshake Token [3]

By scanning the network, the hostname and the port number could be found in the plain text [4]. Through the signature ( to create Ed25519[24] signature algorithm is used), the integrity of the message is protected. After the exchange of the public key, all upcoming messages are encrypted with share secret key using the Diffie-Hellman key algorithm. A handshake token contains a node token and not identity token's hash value. Node token is transmitted in the body field of the message which has the same format as the action message (explained in section 5). The main difference is that the handshake message is transmitted without encryption.

### 5.1.2 Join Message

After successfully exchanging the handshake message a Join message with the token of the identity is sent to verify the handshake message's fingerprint. This token is used to confirm that this identity is running on this particular node. The join message is sent with the identity token. However, the attribute field contains a node fingerprint (nfp), so that Authenticity can be verified by checking the node token's fingerprint of Handshake and Join message. Individual identity does not have node fingerprint in its token.

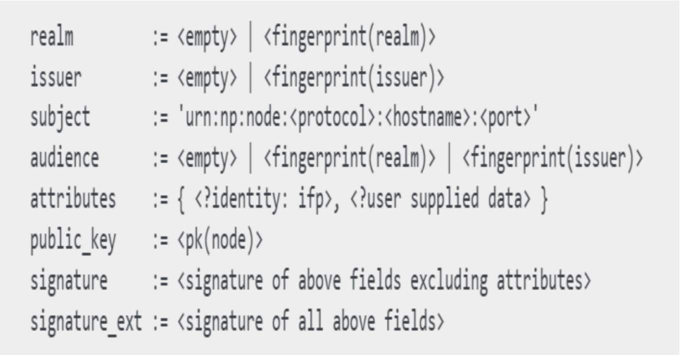
nfp = hash (nodetoken, signature)
ifp =hash (idtoken, signature)



```
realm         := <empty> | <fingerprint(realm)>
issuer        := <empty> | <fingerprint(issuer)>
subject       := 'urn:np:id:'<hash(userdata)>
audience      := <empty> | <fingerprint(realm)> | <fingerprint(issuer)>
attributes    := { _np.partner_fp: nfp, <?user supplied data> }
public_key    := <pk(identity)>
signature     := <signature of above fields excluding attributes>
signature_ext := <signature of all above fields>
```

Fig 4 : Join Message with fingerprint of node [3]

Since one node can have more than one identity, Join message contains also node token with fingerprint of identity instead of node fingerprint(In current situation: one node is connected with one device).



```
realm         := <empty> | <fingerprint(realm)>
issuer        := <empty> | <fingerprint(issuer)>
subject       := 'urn:np:node:<protocol>:<hostname>:<port>'
audience      := <empty> | <fingerprint(realm)> | <fingerprint(issuer)>
attributes    := { <?identity: ifp>, <?user supplied data> }
public_key    := <pk(node)>
signature     := <signature of above fields excluding attributes>
signature_ext := <signature of all above fields>
```

Fig 5:Join Message with fingerprint of Identity token[3]

### 5.1.3 DHT Message

In general, to discover the host ID, the client sends a DNS query to DNS server. In Neuropil, the Domain Name System is not implemented because the DNS server responds to a DNS query that does not contain any information about the initiator except for the IP address. Because of containing only IP address attacker takes this as an advantage and manipulate server with this address.[6] To avoid this attack and maintain privacy the next message carries node token . Instead of the IP address Neuropil protocol uses the fingerprint to identify the host. DHT (Distributed hash tables)[21,22] messages are about nodes for example by sending ping message to measure the latency between two hops or by update message to introduce nodes into the network. If a node wants to stop the connection, leave message is used. For continue the communication Acknowledge message is sent that carries uuid of the initial message. It indicates that a certain message exchange needs an explicit confirmation. It is not necessary that acknowledge messages and the initial message use the same route for transmitting.[4]

### 5.1.4 Message intent token

Nodes have authenticated each other by verifying the signature of the handshake token and through the Join message they get a confirmation of the identity and node. Then nodes share pheromone messages that carry bloom filters. As it transmits through nodes ,the attacker can see but not be able to understand. The messages are transported through the hash table, only a fraction of all used message subjects are visible to it [4].
Following attributes-based access control model [17,18] identities authenticate and authorize each other by sending message intent tokens. These tokens establish a connection between node and identity.

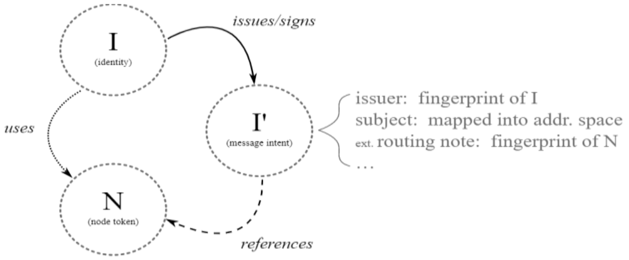
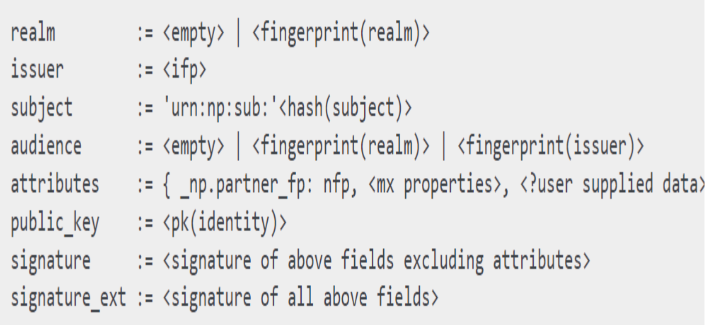


Fig 6: the Connection of message intent token with identity and node token [2]

They are signed by identity so that the receiver matches these to identity and authenticates it and considers it as an issuer. They carry the fingerprint (nfp =hash (node token, signature)) of the node in the attributes field. The fingerprint is the virtual address of a node, so the recipient knows from where the message was sent. The subject field is used to determine the nodes for message exchange. Through the message intent token it is ensured that the peers are authenticated and can communicate in encrypted end-to-end data channels.The header field carries the same subject of Pheromone and payload of body containing sensitive material such as identity token are transmitted in transport layer encryption to authorize only peers[3].



```
realm         := <empty> | <fingerprint(realm)>
issuer        := <ifp>
subject       := 'urn:np:sub:'<hash(subject)>
audience      := <empty> | <fingerprint(realm)> | <fingerprint(issuer)>
attributes    := { _np.partner_fp: nfp, <mx properties>, <?user supplied data>
public_key    := <pk(identity)>
signature     := <signature of above fields excluding attributes>
signature_ext := <signature of all above fields>
```

Fig 7: Message intent token structure[3]

### 5.1.5 Userspace messages

Before sending confidential messages, the Neuropil cybersecurity mesh sends out pheromone messages that carry a “scent” consisting of a bloom filter based on the hash of the message subject. It transmits all possible ways then sender and receiver are also informed about the shortest path to avoid high latency.

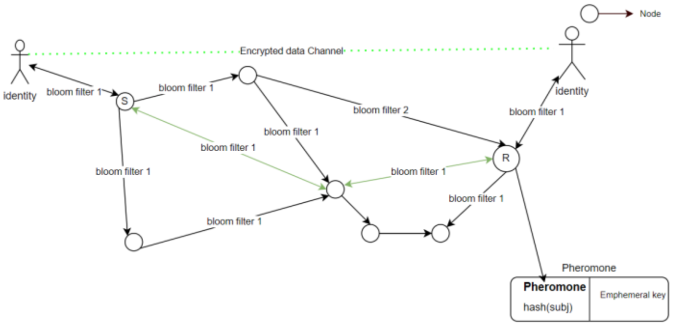


Fig 8: transmission of bloom filter to find match of peer

If a bloom filter can detect a match with a peer , The intent message is sent out containing an identity token referring node token to establish an end-to-end encrypted data channel. An ephemeral key (random value) is transmitted using transport layer encryption plus another share secret key of identities (using DHKE algorithm).The userspace message carries the hash value of the subject, other attributes and is encrypted with DH key and the ephemeral key. With this userspace message the encrypted data channel is established between two identities if the receiver has interest in the same.

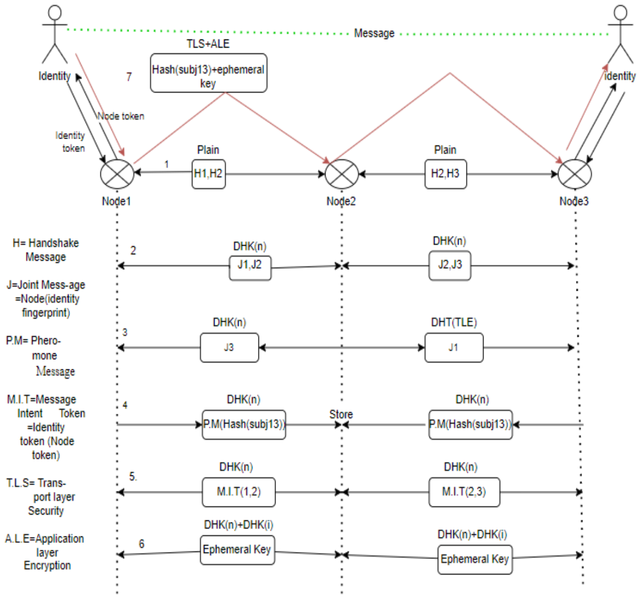


Fig 9:  Data transmission processing in Neuropil Protocol

### 5.2 Define the Security Object

For every message, encrypted data channel is established between two authenticated identities. Neuropil always maintains the privacy and security of entity from creating token to data transmission. Data is transmitted by adding two layers of encryption to communication channel.[2] By using Transport layer Security, eavesdropping attack could be avoided . Neuropil maintains security object of transmitting data which is given below.

**Confidentiality**: Data confidentiality is ensured through the encryption mechanism. Except handshake, messages are encrypted using DHK. After establishing end-to-end data channel, data's encryption has been more strong due to use DHK plus ephemeral key . Consequently, only authorized identity can encrypt and decrypt the message and when the hash function is used, data remains secret.

**Integrity**: Integrity is very important to verify user authorization and data accuracy .As data use TLS protocol which helps to prevent eavesdropping and tampering. For encryption AEAD formula is applied where ciphertext is transmitted with message authentication code (Mac). Stream Cipher is encrypted using ChaCha20-Poly1305[23] algorithm. MAC(n) is formed with token, fingerprint, nonce, key, mac(i) and MAC(i) is with confidential message, nonce and key.MAC will be broken if MITM attacker modify the message. Receiver will understand by comparing the payload with MAC and can detect modifications then drop the message.

**Authentication**: Authentication is one of core elements in Security objects for communication. Before communication begins, identity must authenticate itself, which is done by a signature and fingerprint. In addition, a token contains uuid, public key, signature and fingerprint which are also expression of authentication. Signature is formed with token and asymmetric key . Each token and message contains a signature to authenticate the identity or system.

**Authorization** : End-To-End encrypted data channels will be formed after verifying the authentication. The authentication verification is accomplished by checking signatures with a public key and subject. The identity is authorized to access the network or system. Identities should be authorized before sending the message otherwise a leave message is sent by DHT to terminate the communication. In addition, the nodes check the behavior of their peers and can terminate the connection if peers show the sign of misbehavior.

**Accounting** : Accounting is used to track the activities of an identity when accessing the network or system. It also specifies the amount of data transferred by users and the duration of network access [5]. In Neuropil, the activities of each device are stored over time so that the performance and actions of the authenticated device can be tracked, which is helpful in identifying unauthorized access.

**Freshness** : Another security object is Freshness which can be determined by the following factors. The quality of being able to detect and anticipate replay attacks on a protocol. freshness of messages is a key issue in entity authentication within security protocols, to prevent replay attacks.[2]. Neuropil always preserves the freshness of message. Each message with subject is transferred to a dedicated data channel with uuid and validation time and nonce. After a certain time, a new data channel is established between two authenticated device for a new subject.

#### 5.2.1 Define the Security Object of one Message

**Action Message**

After completing authentication over an encrypted data channel, the client sends a request to the coordinator and can access the device. Through the action message, the client sends the direction in which to move the body parts of the device. Here we select an action message as a representative of all messages and explain its structure and security objects. The encrypted action message is transmitted over an end to-end encrypted data channel, the general format of which is shown in Figure 9. In this ciphertext, only MAC and nonce are sent in plain text and the other parts are encrypted with an symmetric key. With Every transport layer encryption, Instructions field are placed, which defines the sender via a sequence number. In order to make the information more confidential, other parts are encrypted again with an ephemeral key.

After getting MAC and nonce by decoding, the Diffie-Helman key (created from the other public key and own private key) is used to decrypt, the instructions, header and cipher text are found. The next E2E decryption is done with an ephemeral key, and we receive the sender's information. For example, MAC contains an identity information, and the message of MAC(i) is created from header, attribute, body and nonce. The Sender and recipient ID, sequence number of the message, uuid of the message and subject are in the header field. Through the attributes and payload in Body, the actual information is revealed.

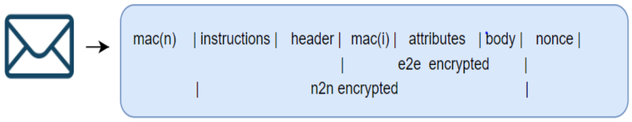

mac(n) | instructions | header | mac(i) | attributes | body | nonce |
| e2e encrypted |
| n2n encrypted |

Fig 10 : The structure of encrypted message[5]

MAC(n) Using for node authentication (DHK+ Message(Instruction+header+e2e encryption+ nonce))

Instructions Sender ID sequence number (adding ID node to node)

Header Subject : ID of the message subject

TTL : Time to live for the message (in seconds)

To : ID of the receiver (fingerprint)

From : ID of the sending node(fingerprint)

Parts : Current/total number of message parts

Seq : Sender ID sequence number

UUID : Unique ID for each message

MAC(i) Using for identity authentication (Ephemeral key+ Message(header+attribute+ body))

Attributes Extra attributes of a message (similar to http header fields)

Body Payload

Nonce Containing random value

E2E End -To-End Encryption from mac(i) to body using ephemeral key (random value)

N2N Data is encrypted using DHK

**Security Structure**

Before sending an action message from client to robot , the message intent token is transmitted first in transport layer encryption for establishing data channels to ensure the identities authentication. By verifying the signature these identities are authorized to access the dedicated channel. As a result, robots can react to a client's demand. The encryption and decryption process of messages is carried out with an symmetric key. Neuropil encrypts the message using the Diffie-Hellman key and an ephemeral key containing random values. Each ephemeral key is unique compared to others .The algorithm used for message encryption, decryption and authentication are given below:

| **Data** | **Security Object** | **Mechanism** |
|---|---|---|
| Status | Confidentiality | TLS-1.3 ,data transmitted end-to-end encryption data channel |
| | Integrity | AEAD (MAC and cipher text which is created using ChaCha20-Poly1305algorithm ) |
| | Authentication/ Authorization | Digital Signature[24]/token, Fingerprint |
| Request | Integrity | Similar to Status |
| | Authorization | |
| Permission Message | Confidentiality | Similar to Status |
| | Authentication | |
| Action Message | Confidential | Similar to Status |
| | Integrity | |
| | Authentication/<br>Authorization | |
| | Freshness | Data validation(ttl,uuid),nonce |
| | Non-repudiation | For Authentication and authorization and dedicated data channel and activity(example movement of Robot according the action message) |

Table 2: Security object of Message with Mechanism

## VI. Attack

Attack implementation defines the performance and security level of the system. Most protocols which are used are familiar and standardized. For that reason, attacks are launched for exploiting security holes in those protocols [7].

In this project, the output of the system comes out through the robot's current status where it mentions authorized client and co-ordinator ID and accessing body parts. So other identities are also informed about co-ordinator ID, client ID who can access the robot, which body parts are currently moving. Through the robot's status the information is spread out among authentic and authorized identities. Data is sent over the end-to-end encrypted data channel, ensuring that unauthorized entities cannot modify the sensitive information during data transmission over the Internet. Without authentication attack is not possible.

In some cases if (for example **MITM attack**) an attacker is able to modify the message in handshake using its own public key, he will not be successful due to signature. Since signature contains node token, it will be broken for any change in token. Consequently Fingerprint will be broken because of signature. But through distributed hash table nodes learn the fingerprint of others.

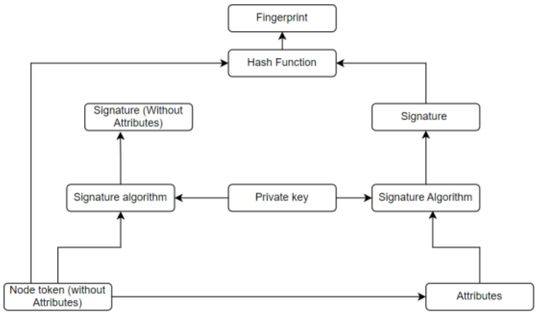


Fig11 : Fingerprint Formation

In following steps the attacker can not able to continue because Join message carries node token and identity token. In Join message node fingerprint is based on node token and references identity fingerprint, similarly identity fingerprint with node fingerprint. It is quite difficult for attacker to break authentic-cation.

As Robot's movement depends on client and co-ordinator, our goal of this attack is to gain access control over the robot or to create a barrier for other authorized clients to access the device. Following this target the attack is applied.

### 6.1 Denial of Service Attack

DOS attack is an attempt to make a device or network resource unavailable to its intended users [8]. Usually, its implementation occurs by sending a large number of request packets which prevent legitimate network users from accessing services or resources of targeted identity. Sometimes these requests come from different locations where a number of devices in the botnet [9] are controlled by hackers. Some the request containing spoofed IP address of sender are sent to a specific port on the target system (called UDP flood attack [10]) and generate a flood of request for target device.

The motive this attack is to prevent other traffic to access the resource or service and after a time making the identity or system shut down.

### 6.1.1 Frame work Of Experiment

In this paper, the attacker keeps the robot in the target so that other identities except the hacker cannot access it. As an individual identity, attackers require authentication. After authentication, it can access the network and send the message to other identities. After establishing the connection between two identities, the attacker sends the request over the data channel. The framework of the experiment is shown in the figure 11.

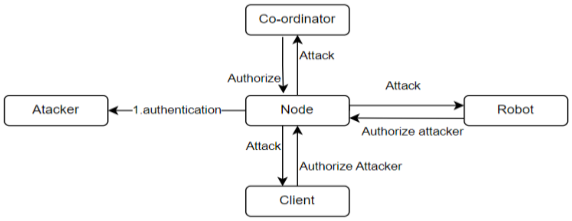
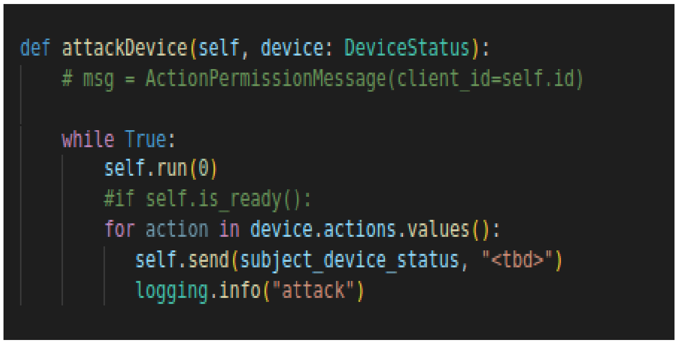


Fig 12 : Frame work of Attack[11]

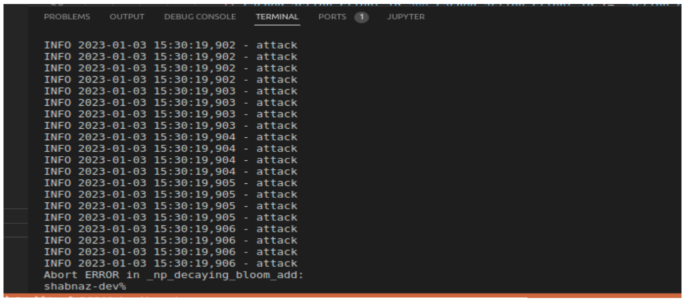


```python
def attackDevice(self, device: DeviceStatus):
    # msg = ActionPermissionMessage(client_id=self.id)

    while True:
        self.run(0)
        #if self.is_ready():
        for action in device.actions.values():
            self.send(subject_device_status, "<tbd>")
            logging.info("attack")
```

Fig 13: python coding DoS attacker

According to the attack , the attacker sends a large number of request for a certain amount of time.

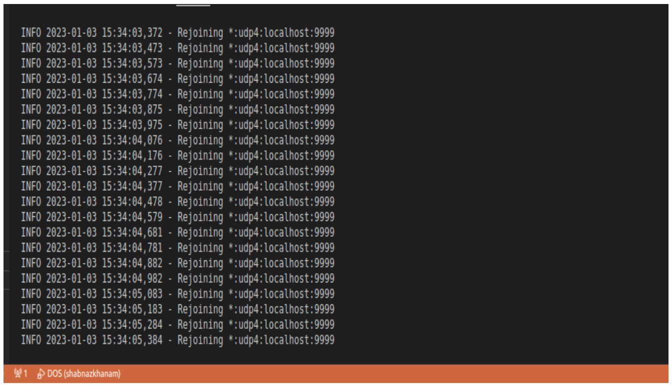


```
INFO 2023-01-03 15:30:19,902 - attack
INFO 2023-01-03 15:30:19,902 - attack
INFO 2023-01-03 15:30:19,902 - attack
INFO 2023-01-03 15:30:19,902 - attack
INFO 2023-01-03 15:30:19,903 - attack
INFO 2023-01-03 15:30:19,903 - attack
INFO 2023-01-03 15:30:19,903 - attack
INFO 2023-01-03 15:30:19,903 - attack
INFO 2023-01-03 15:30:19,904 - attack
INFO 2023-01-03 15:30:19,904 - attack
INFO 2023-01-03 15:30:19,904 - attack
INFO 2023-01-03 15:30:19,904 - attack
INFO 2023-01-03 15:30:19,905 - attack
INFO 2023-01-03 15:30:19,905 - attack
INFO 2023-01-03 15:30:19,905 - attack
INFO 2023-01-03 15:30:19,905 - attack
INFO 2023-01-03 15:30:19,906 - attack
INFO 2023-01-03 15:30:19,906 - attack
INFO 2023-01-03 15:30:19,906 - attack
INFO 2023-01-03 15:30:19,906 - attack
Abort ERROR in _np_decaying_bloom_add:
shabnaz-dev%
```

Fig 14: Sending a large number of request

### 6.1.2 Output

Logically, the robot will not respond to the attacker's request because it only accepts authenticated clients whose ID it receives from the coordinator. But indirectly The attackers succeeded in disabling the robot . Because the robot stores the status of the authenticated identity. These packets of the attacker status became a large size containing random values that might exceed the storage capacity of the robot. The Robot terminates the communication and stops working.



```
INFO 2023-01-03 15:34:03,372 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:03,473 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:03,573 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:03,674 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:03,774 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:03,875 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:03,975 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,076 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,176 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,277 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,377 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,478 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,579 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,681 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,781 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,882 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:04,982 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:05,083 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:05,183 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:05,284 - Rejoining *:udp4:localhost:9999
INFO 2023-01-03 15:34:05,384 - Rejoining *:udp4:localhost:9999
```

Fig 15: Robot disconnect the communication

### 6.1.3 Overcome of this Attack

Attacker's request is transmitted by using Bloom Filters similar to other messages .BF (Bloom Filter) verifies data by checking the incoming data bits with its stored bit in probabilistic data structure. It returns positive if bits are matched with stored bits, otherwise it returns negative [12]. It takes a very small amount of memory to store a data set [12] and mainly performs with three kinds of operation (i.e., insertion, query and deletion) [13]. To perform query operation is necessary before insertion and deletion operation [13].Bloom Filter takes hash values of data. For security and covering all queries/ message data performs with eight numbers of hash function and these values indicate bit locations of bloom filter array. If the indicated bit locations contain 1, BF returns true positive. That means the query will exit in the bloom filter.

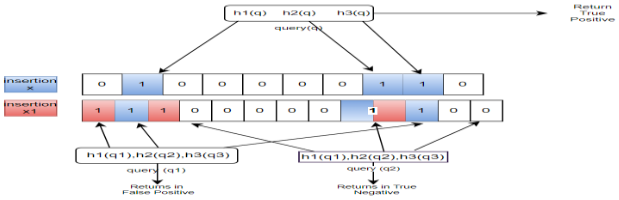


Fig 16: General Overview of Bloom filter

The attacker keeps sending the same message which contains unusual bits. After receiving the message, the robot disconnects the communication. Through the pheromone message node has found the shortest path to communicate then all messages follow the same path to reach the destination. Consequently, the attacker makes the path busy by message flooding and creates a barrier for traffic following the same route. So nodes apply the Decay Bloom Filter (DBF) technique to prevent duplicate messages from being sent.

In DBF, the cell counters are decrement for every insertion and old items are continuously decayed with time. As a result, new items are inserted to support queries making information unclear for older items [14]. Other side Insertion after exceeding threshold value, memory is stopped to insert items [13]. Decaying Message will be proceeded until all bits become zero. For upcoming query BF returns true negative that means they don't exist in bloom filter data array .

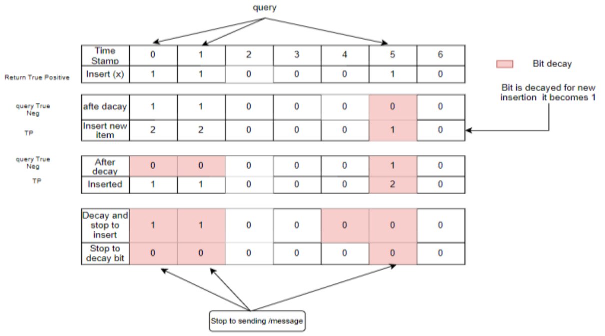


Fig 17: Decay Bloom filter processing

## VII. Discussion

From the beginning of the communication to the termination of the connection, Neuropil preserves all kinds of privacy. For security, all types of cryptographic protocols are used and at each step, verification is maintained by a signature. The secure communication is ensured by using an encryption protocol in the message and data channel. Due to the use of the TLS protocol and AEAD encryption, the data is protected from third-party modification and data validation maintains its freshness. After validation expires, the old data is automatically deleted, which also means that expired data is equally protected from attackers (replay attack). Due to the implementation of all protocols Neuropil is considered to be a strong security protocol. To justify the security, a DOS attack is implemented, but due to the Bloom filter mechanism, DOS attacks can be prevented.

From business perspective, the Robot configuration and maintenance are a lengthy process. In addition if it is affected by adversary and lose its authenticity, it would be more costly in financial and also spreads its effect on the parts of system. To minimize the threat, Neuropil is designed to generate a secure communication between application and system so that unauthorized identity can't get access of Robot. The Robot will make itself shutdown if it gets unnecessary commands. As a result the communication of networks and device will be secured. It is very important to continue the work on the analysis of the Neuropil protocol by using cryptographic tools in the future to verify whether authentication and authorization can be broken.

## References

[1] M.U.R. Mohamed Shibly and B. Garcia de Soto , "Threat Modeling in Construction: An Example of a 3D Concrete Printing System ",, In Proc. Conference Paper, page. 628 Framework of proposed TMM – Quantitative TMM, October 2020.

[2] Pi-lar Gmbh "Neuro:pil Secure Interaction for things" tutorial. https://neuropil.org/tutorial/

[3] Pi-lar Gmbh "Neuro:pil Secure Interaction for things" Protocol Token https://neuropil.org/tutorial/protocol.token.html

[4] Pi-lar Gmbh "Neuro:pil Secure Interaction for things" Security (https://neuropil.org/tutorial/security.html)

[5] Pi-lar Gmbh "Neuro:pil Secure Interaction for things" Protocol message (https://neuropil.org/tutorial/protocol.msg_structure.html)

[6] TinyDNS.org "Advantage And Disadvantages Of Using DNS In Networking"-2018-03-29.

[7] Shio Kumar Singh[1], M P Singh[2] , and D K Singh[3] "A Survey on Network Security and Attack Defense Mechanism For Wireless Sensor Networks" International Journal of Computer Trends and Technology,May to June Issue 2011, section I, pp-1,2011

[8] Mohamed Abomhara and Geir M. Køien "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks", Journal of Cyber Security Vol. 4,Page number-73, 22 May 2015 .

[9] Engr. Umar Iftikhar1, Engr. Kashif Asrar[2], Dr. Maria Waqas[3], Dr' Syed Abbas Ali[4] "BOTNETs: A Network Security Issue", International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 11, No. 11pp-432. (https://www.researchgate.net/publication/347344648_BOTNETs_A_Network_Security_Issue)

[10] Edoardo Biagioni,"Preventing UDP FloodingAmplification Attacks with Weak Authentication", 2019 International Conference on Computing, Networking and Communications (ICNC), IEEE 11April 2019, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8685648

[11] Lulu Liang;Kai Zheng;Qiankun Sheng;Xin Huang,"A Denial of Service Attack Method foran IoT System", 2016 8th International Conference on Information Technology in Medicine and Education (ITME),pp:360-361,IEEE 2019 (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7976501)

[12] Ripon Patgiri1, Sabuzima Nayak, and Samir Kumar Borgohain,"Preventing DDoS using Bloom Filter:A Survey", EAI Endorsed Transactions on scalable information system(Research Article),pp: 1-3,2018. (https://www.researchgate.net/publication/328332457_Preventing_DDoS_using_Bloom_Filter_A_Survey)

[13] Ripon Patgiri;Sabuzima Nayak;Naresh Babu Muppalaneni "Is Bloom Filter a Bad Choice for Security and Privacy?", 2021 International Conference on Information Networking(ICOIN), pp:648-650, IEEE-2021. (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumbe r=9333950&tag=1).

[14] Jonathan L. Dautrich Jr, Chinya V. Ravishankar," Inferential Time-Decaying Bloom Filters" ,AMC Digital library,ACM Trans. Database Syst.,Vol. 44, No. 2,Article 7, pp-239, 2019. (https://openproceedings.org/2013/conf/edbt/DautrichR13a .pdf ) Or https://dl.acm.org/doi/fullHtml/10.1145/3284552

[15] Md Sadek Ferdous, Farida Chowdhury, Madini O.Alassafi, "In Search of Self Sovereign Identity Leveraging Blockchain Technology", IEEE Access,volume7,pp:103063-103070,2019. (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8776589)

[16] Špela Čučko;Šeila Bećirović,Aida Kamišalić,Saša Mrdović,Muhamed Turkanović,"Towards the Classification of Self-Sovereign Identity Properties ", IEEE Access, Volume: 10, 17 August 2022. (https://ieeexplore.ieee.org/document/9858139)

[17] Sheng Ding,Jin Cao, Chen Li,Kai Fan,Hui Li ," A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT", IEEE Access,Volume: 7, pp. 38431 - 38441 , 17 March 2019. (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8668769)

[18] Jialu Hao, Jian Liu, Huimei Wang, Lingshuang Liu, Ming Xian, Xuemin Shen," Efficient Attribute-Based Access Control With Authorized Search in Cloud Storage", ,IEEE Access, Volume: 7,pp.182772 - 182783,21 March 2019. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8672564

[19] Baozhan Chen, Siyuan Qiao, Jie Zhao, Dongqing Liu, Xiaobing Shi, Minzhao Lyu, Haotian Chen;Huimin Lu,Yunkai Zhai, "A Security Awareness and Protection System for 5G Smart Healthcare Based on Zero-Trust Architecture",IEEE Internet of Things Journal,Volume: 8, Issue: 13,pp:10251-10256,01 July 2021. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9273056

[20] Lampis Alevizos,Max Hashem Eiza,Vinh Thong Ta,Qi Shi,Janet Read, "Blockchain-Enabled Intrusion Detection and Prevention System of APTs Within Zero Trust Architecture",IEEE Access,Volume:10,pp. 89271-89272,18 August 2022. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9862967

[21] Klaus Wehrle (RWTH Aachen University · Institute for Communication and Distributed Systems (COMSYS)) ,Stefan Götz, Simon Rieche (University of Tübingen),"Distributed Hash Tables",Conference Paper,January 2005.

https://www.researchgate.net/publication/220978596_7_Distributed_Hash_Tables

[22] Hui Zhang, Student Member, IEEE, Ashish Goel, and Ramesh Govindan,"Improving Lookup Latency in Distributed Hash Table Systems Using Random Sampling", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 13, NO. 5, pp. 1121 - 1123, OCTOBER 2005.

[23] Ronaldo Serrano;Ckristian Duran;Trong-Thuc Hoang;Marco Sarmiento, Akira Tsukamoto, Kuniyasu Suzaki, Cong-Kha Pham, "ChaCha20-Poly1305 Crypto Core Compatible with Transport Layer Security 1.3", 2021 18th International SoC Desig Conference (ISOCC), 2021. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9614016

[24] Svetlin Nakov,PhD ,"Practical Cryptography for developer", Digital Signature-EdDSA and Ed25519, Sofia, November 2018, Software University.
https://cryptobook.nakov.com/digital-signatures/eddsa-and-ed25519

[25] Fabricio E Rodriguez Cesen;Levente Csikor;Carlos Recalde;Christian Esteve Rothenberg;Gergely Pongrácz " Towards Low Latency Industrial Robot Control in Programmable Data Planes ", 2020 6th IEEE Conference on Network Softwarization (NetSoft), PP.01, 12th August 2020.
https://ieeexplore.ieee.org/document/9165531