An Identity based authentication for Internet of Things (IoT)

KIRALE BHARATH ALVA - 18431364, YUVARAJ SRIPATHI – 14677313
CNT5410 - Computer and Network Security
Email: {bharathalva, ysripath}@ufl.edu

Motivation

- Security threats for Internet of Things (IoT) continue to grow.
- To suggest a better alternative to the use of Public Key Infrastructure (PKI) in IoT.
- Reduce the vulnerabilities due to the use of Public key crypto operations.
- Public key crypto operations are not intended for raw data encryption.
- To optimize crypto operations using Symmetric key encryption/decryption.

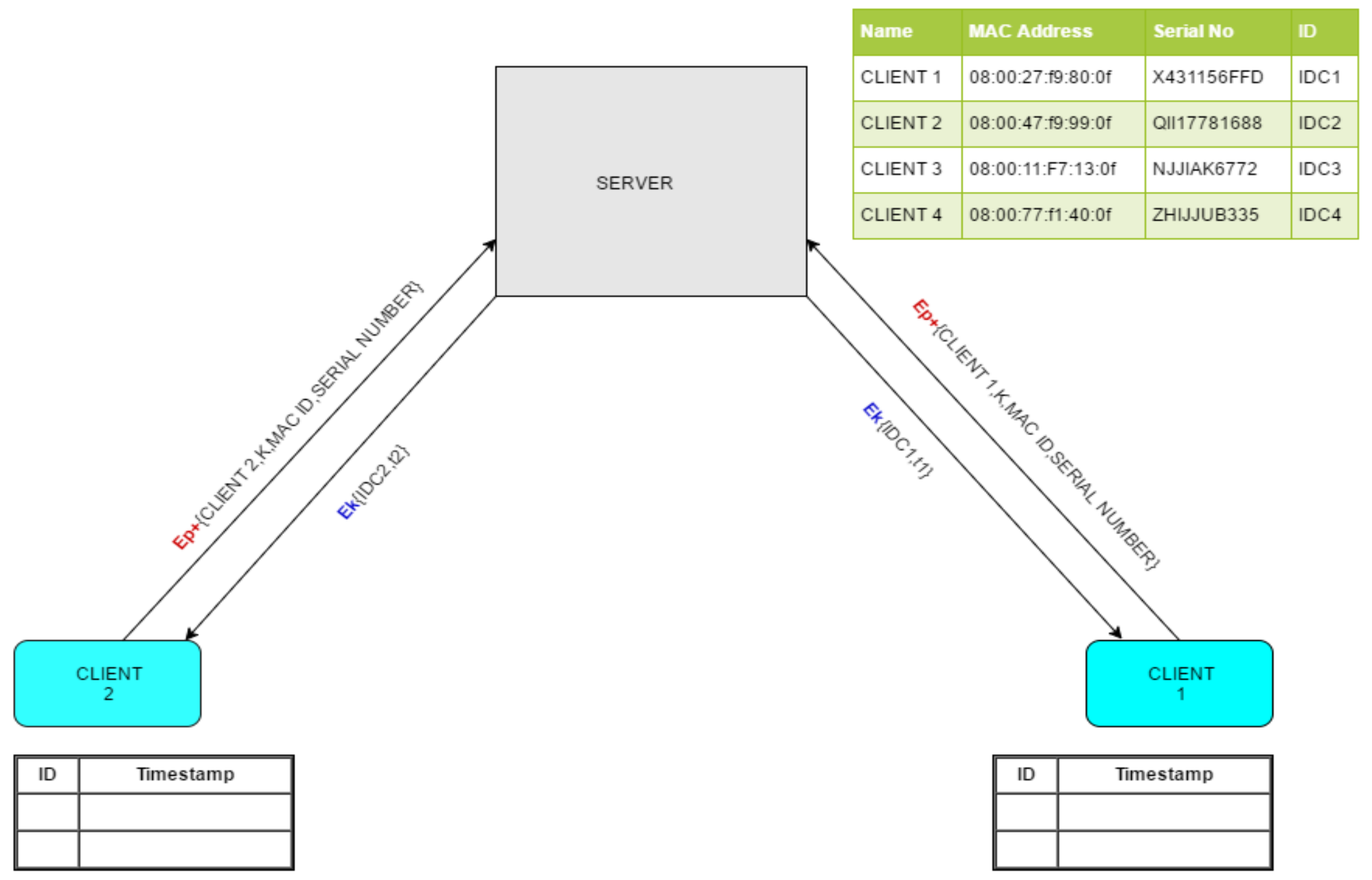
Objectives

- Implement a novel Identity based authentication scheme for IoT.
- One time Identity value used for Authentication and symmetric key crypto operations.

Design

- MAC address and serial number supplied by the Client to the Server along with Client name and a randomly generated key `K' all encrypted in Public Key of the Server.
- Random identity is generated by the Server using MAC address and serial number.
- Identity sent back to the Client encrypted in the key `K' provided.
- Timestamp that indicates the validity of the Identity is included .
- The Server provides the Client with a new Identity once it is expired.
- The Server stores Name, MAC address, serial number and Identity of each Client.

Design

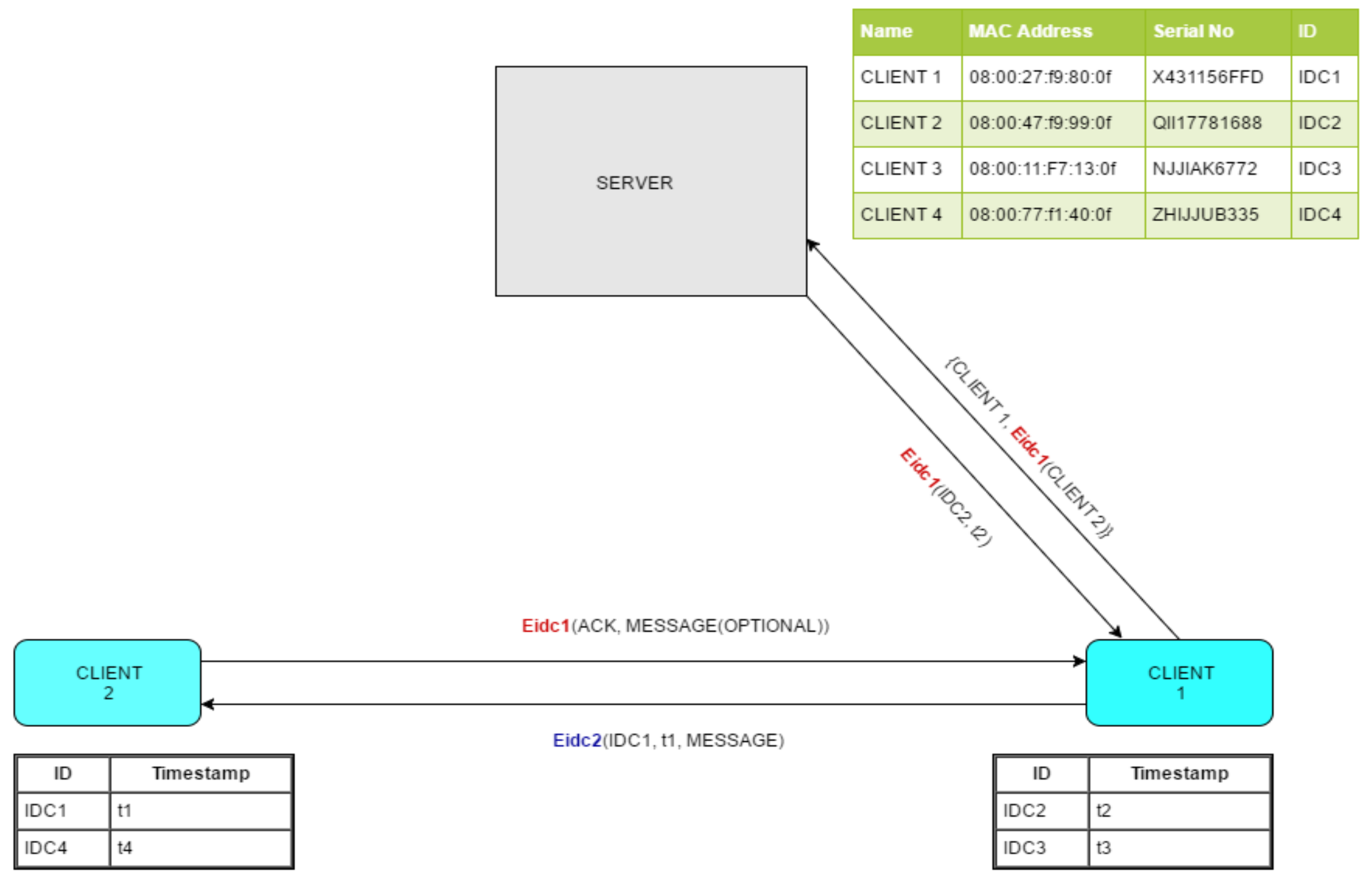

SERVER
Ep+{CLIENT 2,K,MAC ID,SERIAL NUMBER}
Ek{IDC2,t2}
Ep+{CLIENT 1,K,MAC ID,SERIAL NUMBER}
Ek{IDC1,t1}
CLIENT 2
CLIENT 1

Name	MAC Address	Serial No	ID
CLIENT 1	08:00:27:f9:80:0f	X431156FFD	IDC1
CLIENT 2	08:00:47:f9:99:0f	QII17781688	IDC2
CLIENT 3	08:00:11:F7:13:0f	NJJIAK6772	IDC3
CLIENT 4	08:00:77:f1:40:0f	ZHIJJUB335	IDC4

ID	Timestamp

ID	Timestamp

- For communication, Client 1 requests Identity of Client 2 from the server.
- The request contains the requesting Client's name along with the name of the Client whose Identity is being requested which is encrypted in the requesting Client's Identity.
- The server encrypts the requested Client's Identity in the Identity of the requesting Client along with a Timestamp.
- To send a message to Client 2,Client 1 sends it's Identity, message and timestamp encrypted in the Identity of Client 2. An acknowledgment is sent by Client 2 encrypted in Client 1's Identity.
- Timestamp and Identity of communicating clients are saved in the corresponding client's table.



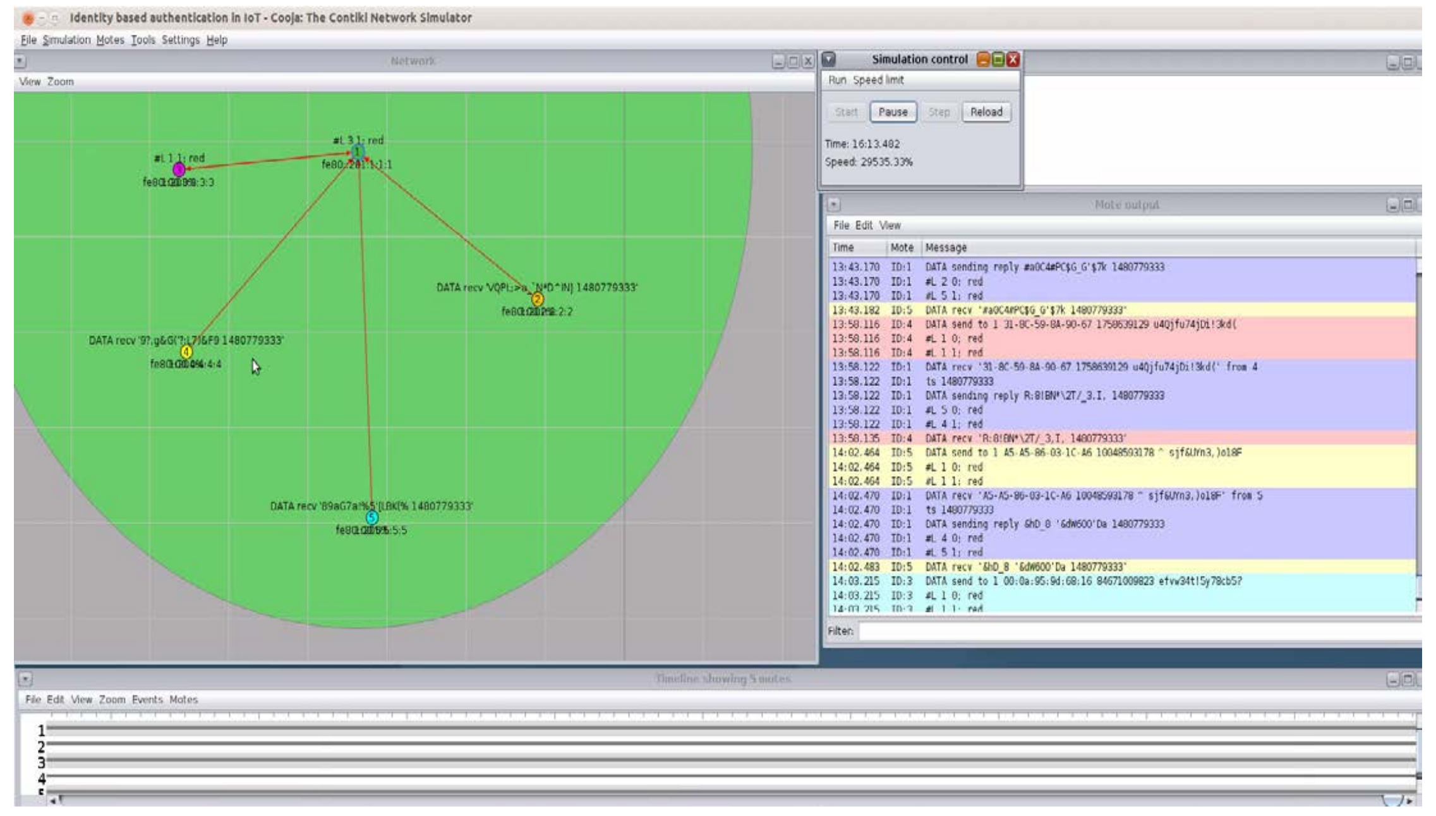
Name	MAC Address	Serial No	ID
CLIENT 1	08:00:27:f9:80:0f	X431156FFD	IDC1
CLIENT 2	08:00:47:f9:99:0f	QII17781688	IDC2
CLIENT 3	08:00:11:F7:13:0f	NJJIAK6772	IDC3
CLIENT 4	08:00:77:f1:40:0f	ZHIJJUB335	IDC4

ID	Timestamp
IDC1	t1
IDC4	t4

ID	Timestamp
IDC2	t2
IDC3	t3

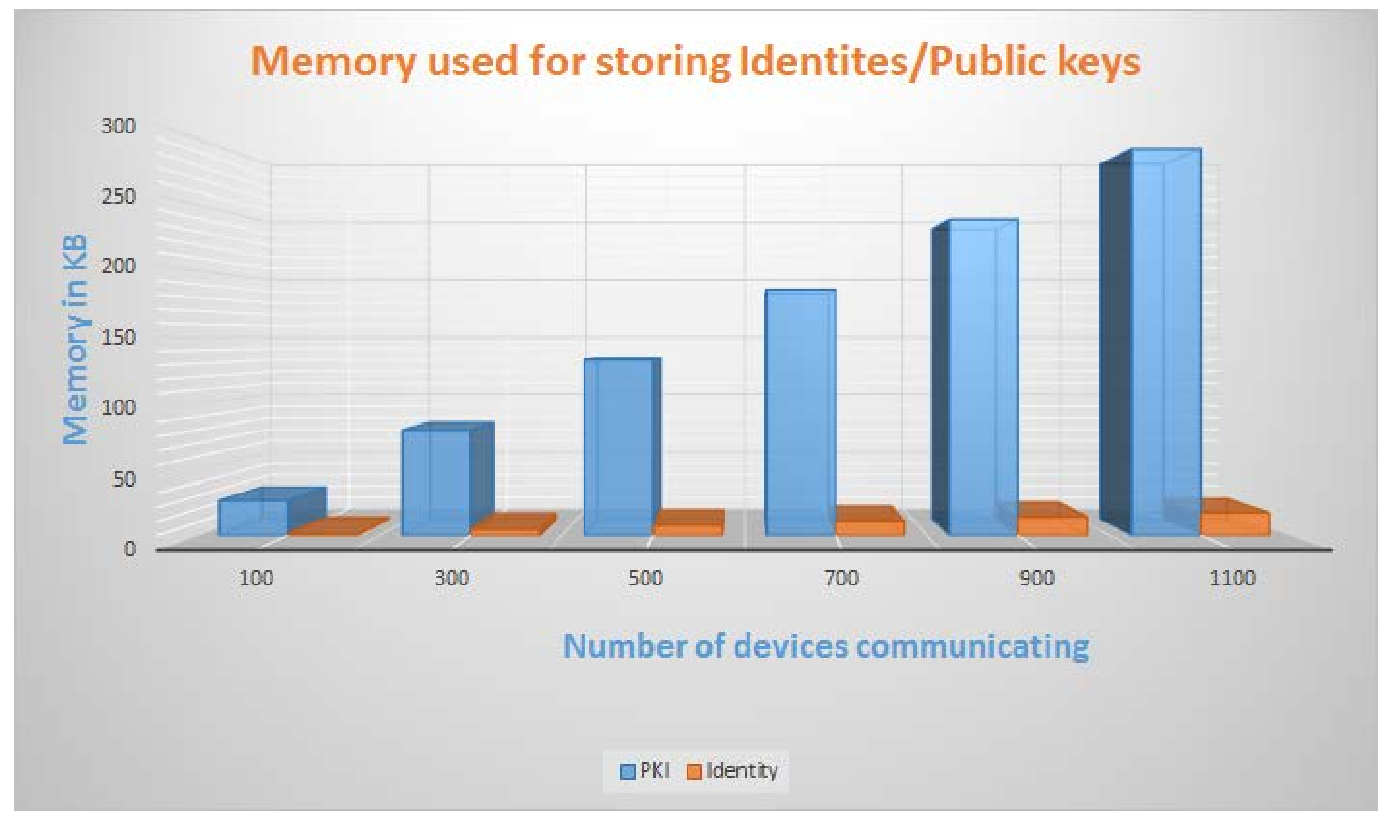
Implementation

- Broadcast of IP address from the Server to all the nearby nodes.
- Establishment of UDP socket connection between the communicating nodes.
- A key 'K' generated by the random generator function to provide the server to encrypt the generated Identity and send it back.
- Secure initial communication between the end device and the Server using the Server's public key.
- Custom algorithm for Identity generation with MAC address and serial number as the seed at the Server.
- Encrypt the generated Identity using key `K' provided by end device which requested for the Identity.
- Encryption/decryption in the Client and the Server is done using AES-128bit symmetric key algorithm.
- A nonce is introduced here in the form of a Timestamp, which indicates the validity of the Identity. Once expired, the Server generates a new Identity and communicates it to the corresponding Client.
- Simulation is done in Contiki Cooja.



Result

- On using an Identity of size 16 bytes and using the same as the key for AES-128 bit symmetric key crypto operations against a public key of size 272 bytes reduces the memory consumed for storing them especially at the central server node and memory constrained end devices in the IoT environment.



- End devices will be able to decrypt the packets that they have received if and only if they are encrypted with its corresponding Identity. This mechanism confirms the authenticity of the device in the other end that wants to communicate with it since Identities are provided only by the server to the authentic devices (with reference to the Server) in the same network.
- The design also provides for resistance against man-in-the-middle and replay attacks.

Conclusion

We have managed to provide a better and robust method of authentication using randomly generated identities for each of the devices in the IoT environment against the usage of PKI by using the same entity for both authentication and encryption/decryption oprtations.