Final Project

Mukesh Dangi

IoT has changed the way we think about smart home. It is changing the way we are interacting and behaving with outside world. Since smart home devices consistently store the data on cloud over the internet, there are many ways by which someone can track home activities, sleeping habits or even diseases by device info on network. One can infer device information just by a DNS query, device purpose, state and traffic rate to check user behavior and data uses even if data is encrypted. IoT system are more susceptible to attack
if Smart home system has insufficient authentication or authorization, unsecure network and cloud platform, poor system configuration.

Thread Model:

Analyzing and documenting the ways each device of smart home could be attacked. Thread model should highlight some key issues such as potential threads to a device, what are most critical assets to protect, thread severity, counter measurements, and how to meet the critical security requirements. There are many ways by which a hacker can gain knowledge about private information of a user without even modifying it. These **passive** attacks are very dangerous because user has no idea that some unauthorized person is eavesdropping because smart home devices are limited in computation capabilities and fully secure protocols are not implemented or monitoring his traffic. Then there could be **active** attacks such as **DOS attack**: in which attacker blocks the access to the authorized user by sending large number of messages to the smart home network, **Identity theft**: used to steal identity for future unauthorized activities in the system. **Replay attack**: attacker sniff the message from sender then send to multiple users breaking the privacy. **User Session theft**: the attacker steals the cookie or token from user session and then performs unauthorized activities. **Malicious Software**: Hacker installs malicious software or malwares on the smart home system to take control of it and play around it in future. Some of the assets we must need to protect are:

- Certificates and device unique keys, Login user name and password
- System configuration to avoid IP spoofing or comptonization
- Voice recording such Google home or Alexa devices
- Event pattern and communication
- Configuration and privileges: tempering config, impersonate, device disconnect
- Device port/channel or resource abuse
- User privacy and potentially life-threatening incidents. Here hacker could overflow the bath tub or disable the fire alarm. So basically, from a little incontinence to a human life, we are putting everything on very dangerous line if we don't build a secure system.
- Kitchen aid Smart: coffee machine, refrigerator, dishwasher, oven
- Personal preferences, voice activated assistants, sleep monitor, Indoor environment Smart: lights, thermostat, air conditioning, windows blinds, floor heating, energy / water meter, solar, home occupation, temperature

Smart home security proposed solution and objectives:

Along with proper functionating of the smart home, it should protect message flow and data integrity. We should mainly focus on following security objectives:

- Confidentiality of data and crypto keys
- Integrity of messages, videos, and other stored information.
- Availability: Every node should be up and ready to take new task without any significant down time. We can use some sort of master slave system architecture for few nodes which are very crucial for smart home like fire alarm, door lock etc.
- Authenticity: Only authorized person should be able to command the system. For example, Alexa should user authorized person to make online order, setup meeting etc.
- Secure life cycle and non-repudiation

Counter Implementation and measurements: Because of Smart home system's limited space and computing power limit, it really hard to build a more secure system.

Authentication and Authorization: Define clear roles and responsibility of each node in the network. Give appropriate access level only. HTTP or any other protocol used in message transmission should use SSL/TLS protocol. Moreover, if we want speedup the development with more security, we should also consider OpenID and OAuth mechanisms. To avoid dictionary or brute force attack we should always use randomness or nonce while generating keys or transmitting the messages. While creating users we should enforce strong password, 2 factor password policy.
System should have separate subsystem designated for identity and service providing. Allocation of privileges should be clearly defined and limit user's capability to collect data from any node.

Secure identity: maintain clear defined roles and auth. Develop trusted communication channel. While accessing from remote, make it's over a secure protocol like HTTPS. While developing the system we need to make sure that we have set threshold limit of the system failure.
Secure Audit: Develop a real time logging system which prints all exception or authorization access. Sound an alarm when someone is trying to access more than a threshold.

Secure Boot and upgrade: Develop interface to boot and upgrade. Maintain an option for current patch update. System should not allow to roll back if the current system was migrated successfully.

Secure Data storage: We need to have data, channels, devices in end to end encryption supported. Most of smart home data should be stored on a sensitive data storage unit with encryption like AES encryption. While communicating to other node, sender node should first implement a handshake and protocol exchange agreement along with message encryption mechanism like RSA where by choosing secret prime number keys a and b hacker would have difficult time guessing the secrete key. Store keys on very secure storage as we all know that secrecy of cryptography is hiding or securing the keys.

No target ads: Smart home system providers should not use any personal information to target ads as per user internet history, behaviour and personal data.

No Data Processing by vendor and some constant listening functionalities should be disabled by default. Within the network, nodes should not include user identification or other crucial information. They would work based token system which was initially created.

Gateway implementation: No node in smart home should connect to internet unless given access. Moreover, node connect to internet via gateway where we have secure channel.
Use Elliptic curve ecosystem for Public key infrastructure.

Since in smart home system, we are not concerning about source and destination anonymity rather securing key and messages. So, using Tor, Onion or zero knowledge system would not make sense. However, since we are focusing more on channel and data security, we should more think about public key cryptography. If we want to challenge our self we can use modified version of block chain for data integrity but any other secure hashing mechanism should be sufficed enough such has HMAC.
Since, communication between two devices could easily eavesdropped we should use any key sharing also which is really hard break and one such algorithm is Diff-Hellman where finding secret keys (a, b) is really hard if attacker eavesdrops and gets pow (g, a*b) so in order to authenticate different devices, we can go for Diff-Hellman key agreement. Some of the nodes like touch screen smart lock or retina scanning devices would need identity-based encryption mechanism.

System Design and Components of Smart Home System:

1. **Smart Home Cloud Hub:** A central powerful ring of machines which takes care of the major computation, message flow, user authentication, device authorisation and addition, secure key storage. We use a ring data structure because one server might not be able to handle many requests and logging activities from sensors, user requests, device communications. This ring follows constant hashing algorithm where each request has a request ID and we hash that ID to get the machine ID to serve the request. This mechanism allows us to distribute the request and response flow optimally. Moreover this central hub could be used for Key distribution KDC as well.

   Data Storage: Personal or device preferences. Store locally on the ring servers so that data doesn't leave outside. Store on third party cloud if data is encrypted using user's private key.

   Key Management: Master key of itself, public keys of all other devices or endpoints.

   Privilege management: it should all to-from messages to ensure that unauthorised person doesn't get the messages which are not supposed to delivered.

   Data Processing: Data processing and scheduling information

2. **Smart Home devices:** The end points of our smart IoT network. Each device share their keys to smart home hub so that in future if one device wants to talk to another, the requester can directly contact to this hub to request a common key K_{AB} which can be used for future communication purpose. Since storage and computation power of each end point device is limited we want to offload most of the work to the central ring servers. Moreover each should have a session and connection mechanism so that ring

hub can perform other tasks as well. Because we don't every device to request for new keys every time for every new message transfer. Every device should maintain a limited time session for every other registered device with the hub.

We can use LRU approach to store the session of recently communicated devices we don't bother ring servers for sessions. For example Air conditioner most the time talks to the user phone or thermometer than other devices so we should store phone session on the AC local storage.

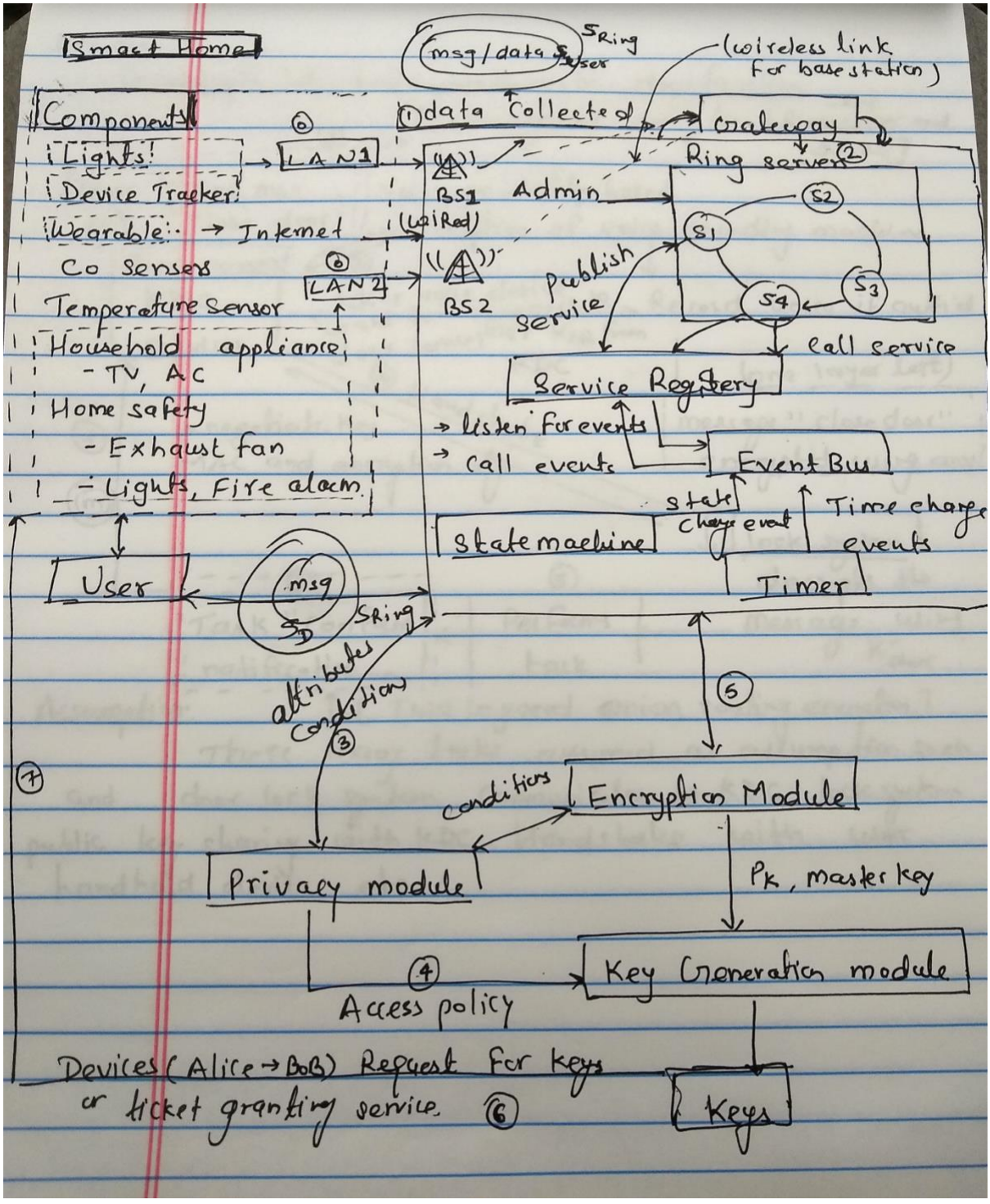
3. **Users:** who uses the services of one or many of the smart home devices. We should have clear defined privileges and roles for every user. User could be admin, guest and regular user. Guest user should not able to modify any setting of any device. Admin should have all the access from device setting to key change to adding new users.

4. **Vendors:** Vendors are not a major participants of the smart home system however, they should have some privileges to test the system, encrypted backup, roll over in case of wrong patches and perform some other authorised activities to run the system smoothly. Vendors should be given least privileged access to the smart home and that on request and proper supervision.

5. **External Service providers:** Any third party network which is needed to perform cloud services of the smart home system. Most of the time we more storages because sensors produced enormous data for example videos recording, or fire alarm or voice recording devices like mobile phones. After some time, we no longer need some of the data for near future use however needed to be store like video recording of house door etc. In this case we some cloud vendor to store the information. Here User should use his private and encrypt those data before uploading them to cloud.

6. **LAN network:** Some of the cases where devices doesn't need to communicate to outside world then it's better to have local area network to save network bandwidth and load on ring server. For example, fire alarm and sprinkler water system could communicate on local area network. Moreover, We can create LAN network based on function of the devices such TV, AC, home safety devices could on one network, temperature sensors, lights fire alarm could be on other network. Since wearable device constantly moving with human or person or animal, they should directly communicate to home system network gateway, cannot be grouped in one LAN.

7. **KDC or Session manager (**part of ring server): Session manger or KDC should give priority to device traffic while handling requests in traffic shaper system. This bucket system has two input buckets of requests, low priority random traffic to obfuscate the device behaviour and high priority bucket of true device messages. Low priority random traffic is later discarded because there is no recipient for that. Every device requests a sessions keys from KDC for long term communication and makes short time connections with relevant devices to send and receive messages.


Smart Home
msg/data
SRing
(wireless link for base station)
Components
Lights
Device Tracker
Wearable
Co sensors
Temperature Sensor
Household appliance
- TV, AC
Home safety
- Exhaust fan
- Lights, Fire alarm
LAN1
LAN2
Internet
BS1
BS2
(LoRa)
data collected
Gateway
Ring server
Admin
S1
S2
S3
S4
publish service
call service
Service Registery
listen for events
call events
Event Bus
state change event
Time change events
Statemachine
Timer
User
msg
SRing
attributes
condition
conditions
Encryption Module
Privacy module
Pk, master key
Key Generation module
Access policy
Keys
Devices (Alice → Bob) Request for keys or ticket granting service
1
2
3
4
5
6
7

Encryption and key management Methods:

As our end point devices have low computing and storage power, end to end encryption on multiple channels we'll have to use asymmetric encryption algorithms. Since in symmetric key encryption every device to store and share keys to others we cannot use it. Moreover if key of one node is stolen then keys of all their nodes or devices also exposed which is another good reason to asymmetric encryption methods.

Since we have many public key cryptography methods, we'll use encryption schemes based of following criteria:

- Level of security(Critical data, medium or low security)
- Key size: here we have to take the storage into consideration
- Speed of encryption and decryption

There are two potential candidates for our smart home security

- RSA which is bases on integer factorization
- Elliptic curve cryptosystem(ECC), based on discrete log problem

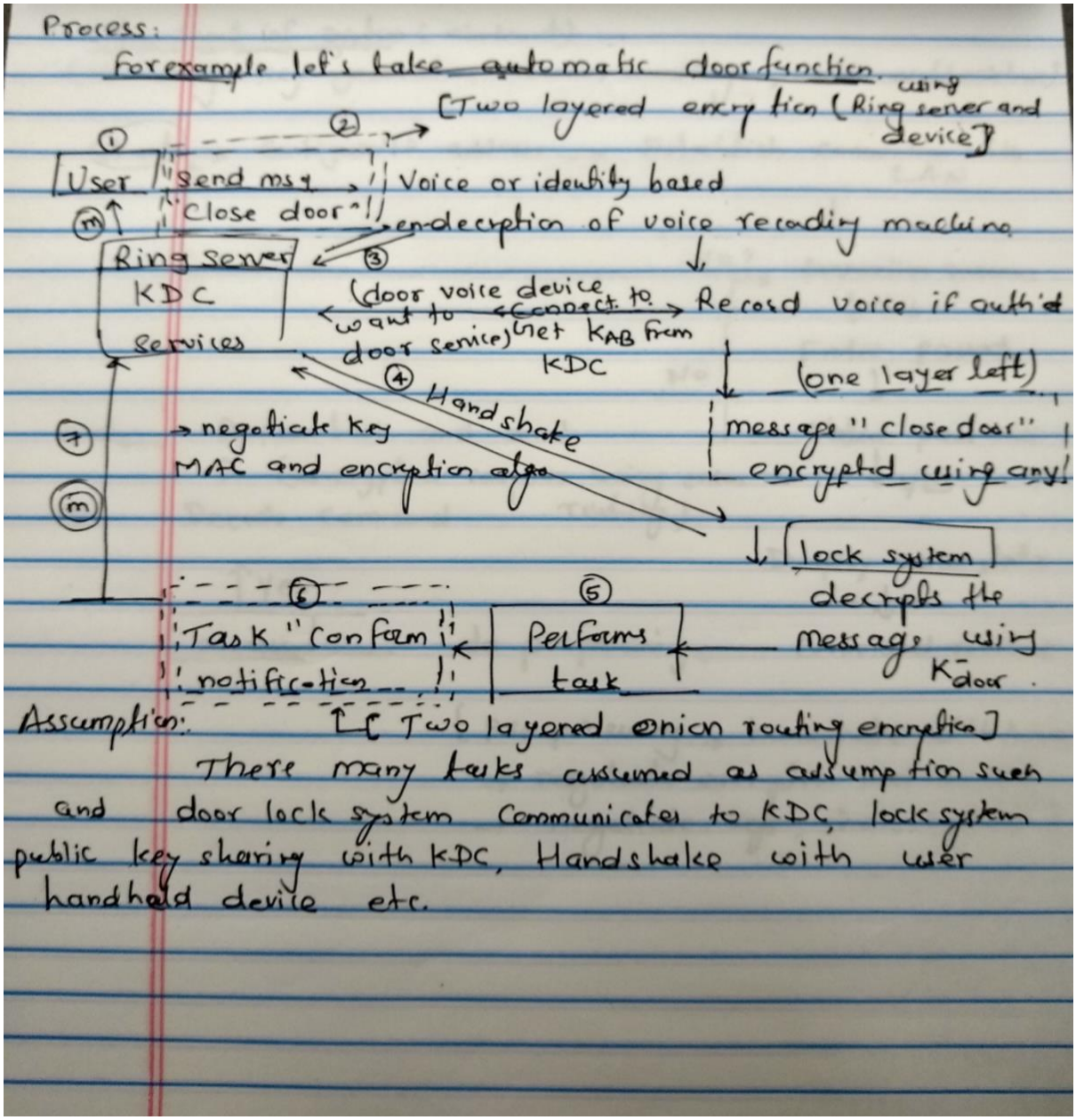
Since in class we studied that ECC key size much lesser than RSA with similar level of security. In our design we'll take 256 bit key length that means attacker has to pow (2, 256) operation to compromise the key.

Since end point devices are limited to computing power we ask ring server to do the most of work of key generation, storage etc. ECC uses Finite Field Arithmetic and the geometry of elliptic curves built on finite fields to create a public key cryptosystem. Let's generate the keys from our ECC and decide where we need to processing-

– Pick a random integer d between 1 and n-1
– Compute Q = d·P, Where d·P is P dot P dot P … d times
– Q = Public Key
– d = Private Key

Finding d from given Q and P is really hard and considered a discrete logarithm problem. The Elliptic Curve Discrete Logarithm Problem is considered hard.
Given d and P, Computing Q=d.P can be done on ring server as this doesn't contain any secret information.

Device Messaging and communication process:



Device Messaging and communication process:

Device communication is susceptible to eavesdrop attack, we need to have end to end encryption, proper routing of messages and handling of sessions to avoid denial of service attack, unauthorised access. Ring server governs whole system's session handling in order to have proper access to the services to authorised parties. Let's take an example where User U wants to use Device D service then

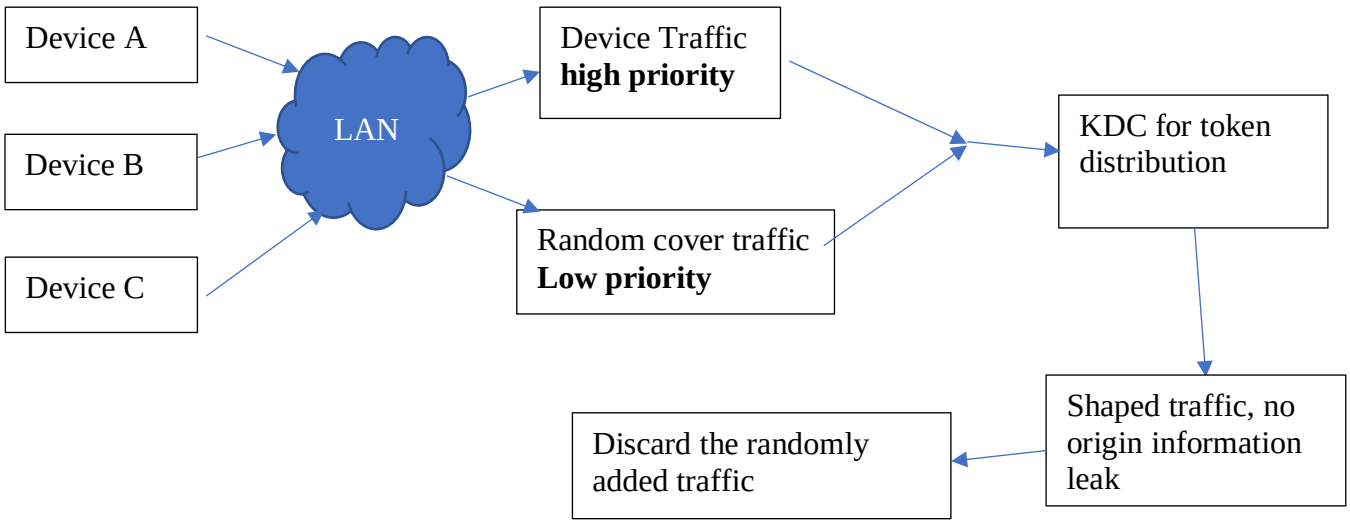
- User U says he wants to talk to device D. Guest users are not allowed to talk to any device in order to make sure that only authorised parties are allowed to create sessions requests
- So ring server R make sure that U is able to sessions ID only if he is authorised.
- Once user U is authorised and authenticated they initiate handshake process to authenticate each other, negotiate encryption & MAC algorithms, negotiate cryptographic keys to be used then use the shared keys for the encryption-decryption and message communication.
- Once both the parties are authenticated, an end to end encrypted channel is setup to secure communication.
- After some inactivity or long session uses, session could be expired to make more secure channel and avoid session hijacking.

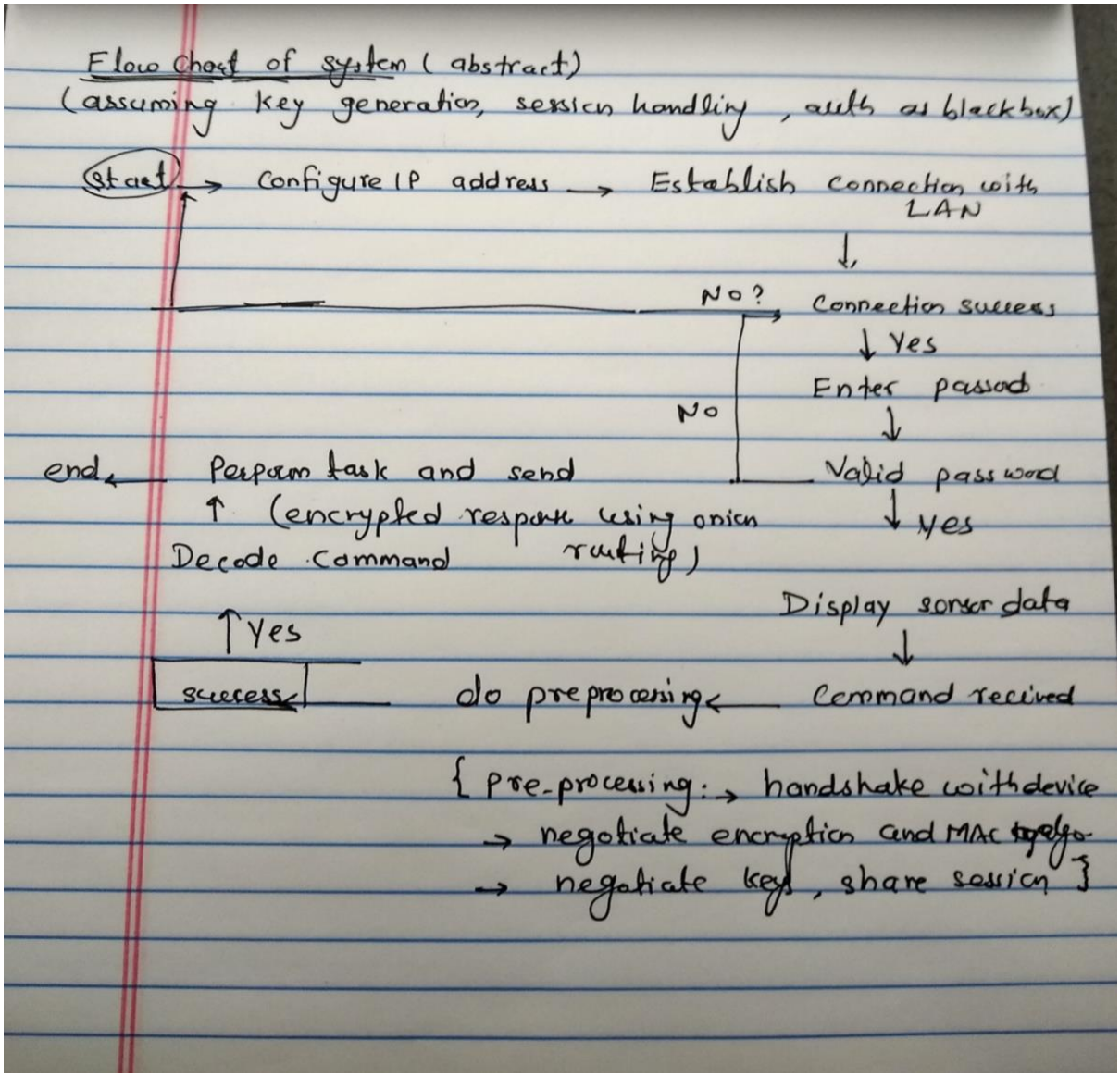
Traffic shaper System design:

One of the practical solution for anonymity of the source or device is adding some random traffic and giving it a low priority so that it doesn't after our device communication. cover traffic could help us in many ways such as avoid device identification and behaviour.

If our data is encrypted and everything is running smoothly, still attacker can figure the device details, device behaviour, working and processing by DNS query, MAC address, traffic rates. For an initial stage we could use difference solution like building a firewall, tunnel traffic but this attacker can get lot of information if there is only single device, sparse devices, dominant devices sending or receiving the messages.

To overcome from the problem of identifying the device and it's behaviour we could implement multiple solutions which we studied in the class. One such solution sending extra random traffic from some devices and then ignoring them.





Routing of messages:

Ring servers plays a key role in initial key generation, session setup and message relay. This design has many advantages like a) every message header required to be check for authenticity of sender. b) there is some sort of anonymity involved because receiver only knows that message came from one particular ring server. Receiver doesn't know the location and identity of sender. Moreover this is really helpful in avoid malware installation and also if one device is hacked, the hacked device has less information about other devices.

Since sender and receiver already negotiated all the encryption & MAC algorithms, cryptographic keys the ring server cannot intercept the messages except forwarding them to receiver end.

We can use onion routing so that one authorised party can decrypt the layer even if someone is able to eve drop on the message. So basically, message sent from user should be encrypted using ring server key and device key and if device sends any notification than messages should

be encrypted using user's public key and ring server key so that only ring server can rely the messages to destination.

Ring server responsibilities are includes but limited to receiving messages from users, devices, authentication, authorisations, session management, open session, notify parties. The ring server has to maintain and allow access the channel according to user or device privileges.

Practical aspect of proposed solution and Satisfying Design requirements:

Over our system should able to perform all the tasks optimally over a secure channel where each end point and user has clear defined privileges and tasks. Let's discuss the satisfaction of initial required condition satisfaction with our proposed system.

Secure Message communication: Our secure key management and message rely requirements meet the need. Ring server opens sessions between two parties. The session is only created when two parties are authorised and authenticated to talk. We use onion routing protocol which also help us to protect the messages as well as correct message recipient so that only sender and receiver know about clear message. In onion routing ring server is able to rely the encrypted messages to correct destination and also able to verify the legitimacy of the sender.

Privacy: Since we are using onion routing, only sender and correct receiver has information about the message rest of the nodes have no information except previous node and next node. Data stored on ring servers can be read by only authorised user or device by their assigned keys. Moreover, ring server architecture is at the home owner premises, third party vendor has no control over the stored data.
Additionally, vendor has no access to any relevant information about the data stored to homeowner's location, vendor cannot process meta data for their profit or any other experiment purpose.

Defending Attacks: Since our ring server analyses all the header information, we can set a threshold which can give as a signal if a hacker is trying to access to rely messages without proper authorisation. Ring server monitors behaviour of the devices and user constantly.

- Man-in-middle attack: all messages are end to end encrypted and uses onion routing and eve cannot guess anything about message however they could be susceptible to timing attack
- Denial of Service: Since we are using ring server as our helper to rely message and maintain sessions for **only** authorised users, ring server has no open session for any irrelevant requester.
- Session Hijacking: Since we are not using any cookie based algorithm and also session replying would not be possible because user-device authentication and limited session time.

Data processing and deletion: only authorised user can delete or modify the data stored at ring servers using his private key.

Complexity of encryption-decryptions:

Let's assume n devices then user will perform 2*n encryptions for onion layer approach, ring server will have n decryptions and each device will have one decryption. Since user devices are good at processing so 2*n encryption should be fine and we are not loading too much work on device which seems feasible and practical design.

Storage Computation and complexity analysis:

Device: since each device need to store it's public-private key pair along with some user public keys for fast processing. Let's say each key is 1024 bits then each device has to store m user keys, 1 of itself and 1 for ring server so in total m+2 keys and total storage at the device end is (m+2)* 1024 bits

User: Since user has to store ID as well keys of the devices and ring server, User should have little more storage and computing power than each end point device. So the total storage is (n+2)*1024 bits and (n+2) * 64 bits of storage. Here we are assuming the we will not have more devices where number device would not cross the 64 bit counter. Moreover, we might need some local storage as temp buffer for message commands and notifications.

Ring Servers: Ring is more powerful and complex system where it stores device id, message, keys, sessions. Ring server stores m users, n devices and 2 keys for itself. Additionally, ring server maintains k sessions, m *m user privileges matrix , n*n device privileges data,

Conclusion:
We presented thread models where mentioned what assets are at risk, proposed a system with detailed component analysis which takes care of data integrity using HMAC or block chain, authentication, & authorisation using session management or KDC/PGP, message routing using onion routing, traffic management & shaper using random traffic generator and session token buckets. Additionally, we explained how our system is able defend itself from outside attacks. At the end also analysed space and time complexity of each component of the system which meets our requirements and could be scaled well on high traffic scenarios.

References:

[1]. Arm Community – five steps for secure IoT building process
[2]. Prof Tanya's Lecture Slides
[3]. Security Analysis of Emerging Smart Home Applications
[4]. Enhancing Privacy in Smart Home Ecosystems Using Cryptographic Primitives and a Decentralized Cloud Entity by R. M. Vrooman
[5]. Deployment of smart home management system at the edge: mechanisms and protocols
[6]. Spying on the Smart Home: Privacy Attacks and Defences on Encrypted IoT Traffic