A Framework for Efficient and Secured Mobility of IoT Devices in Mobile Edge Computing

Sufyan Almajali [+], Haythem Bany Salameh[*], Moussa Ayyash[⋆], and Hany Elgala[◇]
[+]Princess Sumaya University for Technology, Jordan
[*]Yarmouk university, Jordan
[⋆]Chicago State University, USA
[◇]University at Albany-SUNY, USA

***Abstract*— Mobile Edge Computing (MEC) provides an efficient solution for IoT as it brings the cloud services close to the IoT device. This works well for IoT devices with limited mobility. IoT devices that are mobile by nature introduce a set of challenges to the MEC model. Challenges include security and efficiency aspects. Achieving mutual authentication of IoT device with the cloud edge provider is essential to protect from many security threats. Also, the efficiency of data transmission when connecting to a new cloud edge provider requires efficient data mobility among MEC providers or MEC centers. This research paper proposes a new framework that offers a secure and efficient MEC for IoT applications with mobile devices.**

I. Introduction

Internet of Things (IoT) represents an extensive network of devices connected to the Internet. These devices exchange data with each other to provide automation. IoT is expected to make a significant impact on our daily life activities [1]. IoT applications exist in many areas and sectors, examples of such area are Medical and health, Office and business, Home-based and consumer level, Security, Food processing/restaurant, Industrial, Automotive, and Military. Many of the IoT devices and users are mobile by nature. Examples of such mobile users include mobile health monitoring IoT device that is attached to the patient. Also, IoT enabled cars are mobile by nature.

IoT devices vary in terms of size, power needs, and computing capabilities. In general, IoT devices are limited in resources. This imposes security and performance challenges on IoT in general. Achieving security in IoT is important and challenging [2] and personal information should be protected to avoid any unauthorized access.

Traditional IoT applications require continuous data transmission from IoT devices to a centralized data storage location specified by the application owner. Usually, the centralized location is represented by a set of servers that reside at some cloud provider network. In case IoT devices are localized to a certain location, a cloud provider, and an edge provider that is close to that location can be used to implement an efficient solution. However, several IoT applications support mobile IoT users and devices. Examples of such applications are IoT-based health monitoring applications. A patient with an IoT device might move from one location to another. The device movement might mandate network provider change. A network provider change is possible for certain types of IoT device even when residing at the same location such as the ones that use WiFi connectivity option. The security can be achieved when an IoT device talks directly to a cloud provider. Mobile Edge Computing (MEC) providers can be used to improve the efficiency and resources utilization for IoT applications. Besides efficiency issues, achieving security for both MEC providers and IoT users requires new means and mechanisms [3].

The main services used in security are confidentiality, integrity, and availability. Implementing these services will provide a higher level of secured systems [4], and can ensure the safety of information transformation inside IoT. IoT devices exist everywhere in our daily life, any authentication framework in various environments should comply with a suitable IoT standard, to minimize the risk of online users privacy and security.

Recently, IoT has evolved rapidly and has gotten attention in many fields. IoT can involve various types of environments such as monitoring, e-commerce, and smart-house. Furthermore, these services will provide information related to the user and can be integrated into one single system that shares user interface [2].

While these systems and services related to users information, privacy and security aspects, are highly critical, due to the different structure of the IoT. Therefore, the authentication process is considered an important element in the security aspect. Current authentication solutions, do not suit different environments of IoT [2] [5] [6]. Using third-party MEC providers as a middle layer between IoT devices and application centers introduces a set of challenges for supporting mobile IoT devices. We list the following challenges:

- MEC provider selection: How can a mobile IoT device know the right MEC provider to connect to?
- MEC provider selection: How can a mobile IoT device know the right MEC provider to connect to?
- MEC provider authenticity for the IoT application. How can an IoT device verify the identity of the MEC provider?
- Data Confidentiality: How can an IoT device deliver its data to the data center securely without compromising the confidently of its data with the third party MEC providers?
- Data Integrity: How can an IoT device deliver its data to the data center securely without compromising the integrity of its data while passing via third party MEC providers?
- Provider change: How can an IoT device deal with a provider change that comes to a direct consequence of mobility?

978-1-5386-5896-3/18/$31.00 ©2018 IEEE

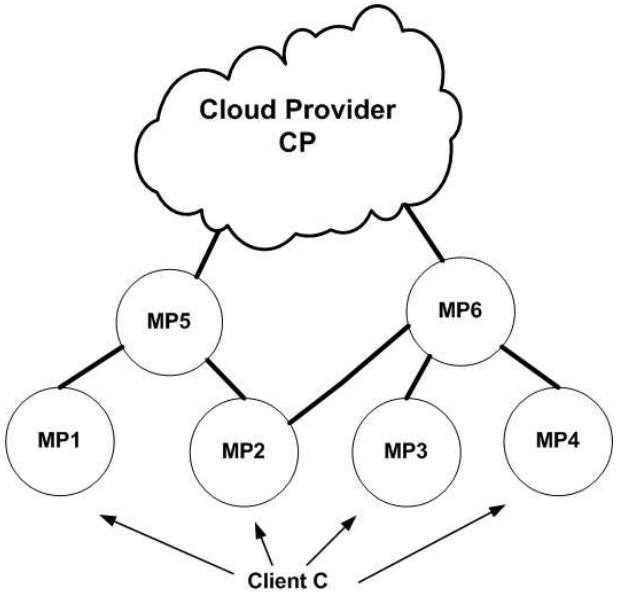

Fig. 1. IoT Client with Cloud and MEC providers

- Efficiency: how can we achieve data confidentiality, data integrity, and device mobility all efficiently via a third-party MEC providers?

This paper proposes a new framework that offers a secured and efficient MEC for IoT applications with mobile devices. We first propose a framework that is efficient and suitable for IoT applications. Then we conduct a comparison among the most known authentication frameworks that can achieve a secure communication framework for IoT to prevent unauthorized users from accessing IoT resources and gaining personal information [2] [5] in traditional clouds.

II. PROPOSED SOLUTION

The proposed solution includes a set of components: The IoT application, IoT devices, Application data center, Cloud Provider, MEC providers, and the solution protocol. We will explain each component using a real-world scenario, a health monitoring IoT-based application.

The IoT application has two parts: a client and a server. The client part is installed on the IoT device. It collects data from the IoT device and sends back to the server part. Also, every IoT application has a set of application requirements (ARs) including security requirement, bandwidth requirements, delay requirements, and cost requirements. In our health-monitoring example, the data could be heart-rate readings and temperature readings of a patient.

The IoT device runs the client part of the application. We target mobile IoT devices. In our health-monitoring example, the device could be a watch reading the heart rate of patients.

The application data center includes a centralized location for storing all of the application data. In our health monitoring application, this could be a cloud provider hosting an application for a hospital. Figure 1 shows the networking model of our solution. An IoT device runs a client (C) application that connects to the cloud provider (CP) via MEC providers MP1, MP2, and so on. A MEC provider could be an Internet Service Provider (ISPs), a Telecom Provider, or a specialized MEC provider. In the Internet world, ISPs are known to be of different tiers. A MEC provider could be Tier 1 ISP, Tier 2 ISP and so on. This implies that the path from a Client C to a cloud provider (CP) can pass through several MEC providers where each one can serve as the middle layer for our proposed solution. When C starts talking to CP, our solution suggests to dynamically select a MPi as the middle layer that will speed up the data transmission, improve its efficiency, and reduce resources consumption for IoT devices. The proposed solution selects the closest MP that achieves the application requirements ARs.

Figure 2 explains the main phases of the solution using a state chart diagram. The first phase includes IoT device registration. This includes registering the device with CP where each device has a Client ID and a password. Registration does not include MPs. IDs and Passwords are stored locally at the device and online at the CP end. Passwords are never communicated as a plain text over the network.

Our proposed solution uses public key infrastructure along with symmetrical encryption protocols to provide a secured transmission to third-party MPs. Every provider has a public digital certificate certified by a third party certificate authority. This implies that every provider whether CP or MP has its own public and private keys while there is no need to have a digital certificate for clients. Table 1 shows a set of parameters used by our proposed framework. The listed parameters are used during the different phases of the framework.

The second phase typically starts when an IoT device attempts to connect with the data center, the CP. The IoT device is represented by the client C. A client C initiates a message to the CP. Figure 3 shows the three main steps happen at this phase.

Table 2 explains the set of messages exchanged at this phase. Each step has one or more messages. In step 2.1, client C sends a message M1 to the CP. M1 contains CID, Cpassword, TS1, and Cnetaddress. The message is encrypted using the CP public key. The message allows the CP to authenticate the client who is requesting to connect to some MP. The time stamp TS1 added is to prevent replay attacks. A message hash added to support the integrity of the message. The message is hashed by the shared key between C and CP. We referred to this in Table 1 as KeyC,CP. We expressed this entire message M1 in the form Enc(M1(CID,Cpassword,TS1, Cnetaddress),PBCP)+Hash(M1). This structure of message formatting repeated in the proposed solution with changes to the content of the message and key used to encrypt the message. Using messages M2 and M3, the CP communicates with one or more MPs to locate the nearest MP that satisfies the application requirements AR. The criteria used for locating the MP includes checking the bandwidth, the security, the delay, and the cost offered by MPs. M2 might be sent to more than one MP until one MP is found. Once the CP identifies the MP, the CP sends message M4 to the MP with a Client network address, temporary Client ID, and temporary password in addition to a secret shared key KeyC,MP to be used by the client C and the MP. This allows the CP to hide the original client ID and passwords from MPs. At the same time, it allows the MP to authenticate clients before communicate using he MP resources. The CP provides the temporary ID and password along with the MP network address to the client C. At this point, C has the complete required information to

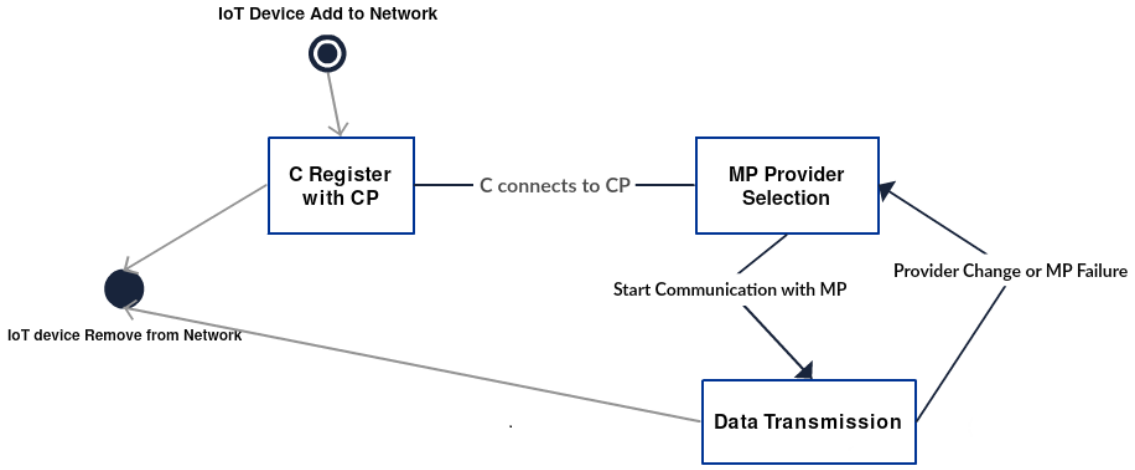

Fig. 2. Solution Phases as State Chart Diagram

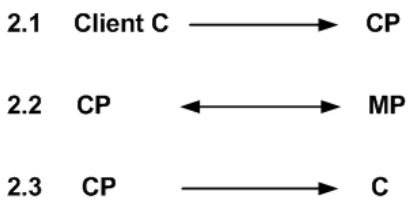

Fig. 3. Phase 2 Steps

initiate the communication to the MP provider.

Next, the data transmission phase starts. At this point, the client C can communicate directly to MP and there is no need for the long communication with CP. This speeds up the communication from IoT device perspective and consumes less power and computing resources. The M6 message is used to authenticate the C to the MP. Once authenticated, the client C can start encrypted data transmission to the MP using M7 or M8 messages. The data transmitted via M7 and M8 is encrypted using a secret key unknown to the MP. The MP accumulates the messages received from C and sends them asynchronously using message M9 format to the CP without involving the client C. Table 3 shows the main messages for this phase.

The client C might change network providers at any point in time due to several reasons such as roaming, switching WiFi networks and so on. This requires that C and CP run the second phase again to select a new MP before continuing the communication. In addition, in case of MP failure or availability issues, the CP can select a new MP. This allows the solution to handle client mobility efficiently and without sacrificing the security of the application. In summary, the proposed solution covers several security aspects of the application. First, it covers confidentiality aspect as it does not reveal the ID of the client, the password of the client, nor the data being transmitted. It supports authentication as IoT clients are being authenticated by MPs without compromising IoT ID info. In addition, the solution covers the integrity aspect by hashing all messages for integrity check. Also, the solution prevents man in the middle attack and replay attacks using time stamps and encryption.

TABLE I
SOLUTION PARAMETERS

Parameter	Value
CP	Cloud Provider
MP	MEC provider
C	Client
$MP_{netaddress}$	MP Network Address
C_{ID}	User ID for client
$C_{password}$	Password for client
$C_{netaddress}$	Client Network Address
$Key_{C,CP}$	A shared Secret Key between C and CP
$Key_{C,MP}$	A shared Secret Key between C and MP
CT_{ID}	Temporary User ID for client known to MP
$CT_{password}$	Temporary password for client known to MP
AR	Application Requirements
$PB_{CP}\&PR_{CP}$	Public and Private keys of CP
$PB_{MP}\&PR_{MP}$	Public and Private keys of MP
TS	Time Stamp
LS	Life Span
$Data$	IoT Application Data

III. SIMULATION RESULTS

We conducted initial simulation experiments to measure the performance of the proposed framework. We used a cloud-specific simulator known as CloudSim. It is a Java-based simulator known in the field of cloud simulation. Our simulation setup studied the performance of the solution when an IoT device talks directly to the cloud vs going through a MEC provider. Table 4 shows the simulation used for each cloud provider as we simulated each role of cloud providers/MEC providers using one data center with two hosts with several virtual machines. The exact specifications are shown in table 4. In Figure 4, we show the time difference between MP based and CP based IoT solution.

IV. RELATED WORK

Authentication process in IoT is necessary to achieve a high-level security and maintain privacy. Next, we present different authentication frameworks with different solutions to achieve security and privacy. Kashif Habib et al [7] discusses supporting authentication for IoT application about monitoring patients outside the hospital. The principal objective of Remote Patient Monitoring RPM system is to use the IoT service to provide a continues monitoring of patients vital signs. The

TABLE II
PHASE 2 MESSAGES

Step	Message	From	To	Message Structure
2.1	M1	C	CP	Enc(M1(C_{ID},$C_{password}$,TS1, $C_{netaddress}$),PB_{CP})+Hash(M1)
2.2	M2	CP	MP	Enc(M2(AR,TS2),PB_{MP})+Hash(M2)
2.2	M3	MP	CP	Enc(M3(Decision,TS3), PB_{CP})+Hash(M3)
2.2	M4	CP	MP	Enc(M4(CT_{ID}, $CT_{password}$, $C_{netaddress}$, , $Key_{C,MP}$, ,TS4,LS), PB_{MP}) + Hash(M4)
2.3	M5	CP	C	Enc(M5(CT_{ID}, $CT_{password}$, $MP_{netaddress}$, PB_{MP} ,TS5,LS), $Key_{C,CP}$)+Hash(M5)

TABLE III
PHASE 3 MESSAGES

Step	Message	From	To	Message Structure
3.1	M6	C	MP	Enc(M6(CT_{ID},$CT_{password}$,TS6, $C_{netaddress}$),PB_{MP})+Hash(M6)
3.2	M7	C	MP	M7(CT_{ID},Enc(Data, $Key_{C,CP}$),TS7)+Hash(M7)
3.2	M8	MP	CP	ENC(M8(CT_{ID},Enc(Data, $Key_{C,CP}$),TS8),Key C,MP)+Hash(M8)
3.3	M9	MP	CP	Enc(M9(CT_{ID},Enc(Data, $Key_{C,CP}$),TS9), PB_{CP})+Hash(M9)

authentication process is crucial in this case. The unauthorized access to patients data will violate their privacy, and intercepting the communication session between the patient and the health system to make a change which could cause a disaster. In this approach [7], it provides dual authentication by combining biometric and radio fingerprint. By using the radio fingerprinting only, the device is authenticated. However, the patient is not. Also, by using the patients biometric only, the patient is authenticated but not the device.

Using biometric techniques in authentication is much stronger when compared to password or token based authentication because its uniquely identifiable among humans. The radio fingerprinting technique uses the signal generated by the wireless device for unique identification. Thus it can be used to identify mobile phones and any other devices that can be used in this case. It can be used to compare different wireless devices. To summarize, the authentication framework includes three phases: Patient to smartphone authentication phase, Smartphone to network authentication phase, and Patient to remote medical server authentication phase. Devices radio fingerprint is checked at the access point to make sure that the traffic is coming from the correct device.

The final phase is to authenticate the patient with the health server. The continuous biometric authentication is achieved in this phase. Patients identification template is called Biometric, the template is extracted by using his

TABLE IV
PROVIDER SPECS

No. Data Centers	1
No. Hosts	2 per Data Center
No. Processors	4 per host 3000$MIPs$ each
RAM per host	8GB
BW	10$Gbps$
Processor Per VM	1 per VM 1000 MIPs
RAM per VM	500MB

Electrocardiography (ECG), or blood pressure features and this template is stored at the health server for future comparison.

Guanglei Zhao et al [7] presents a novel mutual authentication scheme for Internet of Things, which provides an authentication based on a combination of hash function and extraction process. This combination reduces the transmitted information over the wireless network, and the extraction

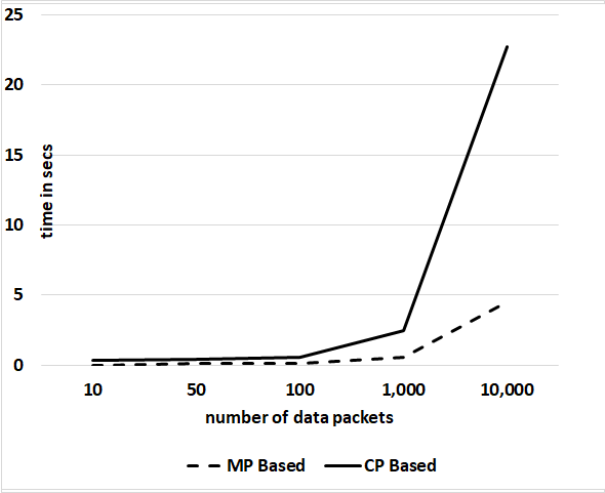

Fig. 4. MP based vs CP based secured data transmission

feature is an irreversible process, if an attacker gets the extracted information, he cannot get the original information, and this ensures the confidentiality. Extraction process means transforming the input data into the set of features to perform the desired task by using a reduced representation instead of the full-size input. This feature can be applied to different environments of wireless networks due to the limited computational abilities and memory resources. Combining the SHA1 and feature extraction in the proposed authentication is the main objective of this research paper. Using this authentication framework improves the security and reduces the network traffic. This technique assumed the use of a Certification Authority center (CA) to verify the certificates of both of the platform and the terminal node. The authentication framework in this research paper is being achieved by completing the following three phases:

1) Initialization Unique ID is being used to identify the terminal node.
2) CA verification - To verify the certificates of both of the platform and the terminal node.
3) utual authentication - Use SHA1 and feature extraction together to achieve the authentication.

In the third phase, SHA1 is being used to convert the authentication information into several messages with fixed

length. Then feature extraction is being used to compute their corresponding variance and energy; they will be transmitted over the network. Tuhin Borgoh et al [8] uses the concept of multi-factor authentication to achieve authentication. In their work, they explained a multi-factor authentication system, the two-step authentications in IoT. In this case, users mobile device is being used to perform the two-factor authentication by using one-time password sent to user‘s mobile through SMS or email (the idea of a token). OAuth 2.0 framework over SASL authentication framework requests from the client authorization. DTLS stands for Datagram Transport Layer Security; a delegation server is used for the initial establishment of the connection from the subsequent application data protection. OAuth is an open standard authorization and authentication framework. It achieves the authentication by establishing an approval interaction between the third-party application and the resource owner. Using OAuth over SASL is achieved by the following phases:

1) Client requests for access to his private resources at the targeted server by authenticating himself with the access token.
2) The server verifies the access token. Then the client is authenticated to access his private resources. Authentication and access control technologies can help IoT users avoid many security issues. Several factors should be considered in IoT infrastructure. Security, scalability, efficiency, effectiveness and the quality of services.

Selecting credentials is the first stage of authentication, these credentials are being used to obtain authenticity in IoT. Moreover, these credentials consist of ID and password pairs, are being used online in the different services in IoT. Other biometric properties techniques to identify the users identity and to obtain the authenticity are Fingerprint scanners, face detection, sound analysis, and iris scanner. These types of authentication techniques require high computation and processing, thus they are not perfect to authenticate in IoT, these systems are usually combined with another authentication mechanism such as password and personal identification number. Objects in the IoT have power and storage constraints, thus, instead of using biometric techniques to identify users authenticating that cause traffic overhead, A Two-Way authentication can be used, it is a process which both entities in communications can authenticate each other. A user can authenticate the server and vice versa. The Two Ways authentication is gaining acceptance as a tool that can minimize the risk of online fraud [1], [5], [9].

The proposed authentication standard in [2] provides access to IoT for authorized users, based on four entities: User, Things, Registration Authority and Head Registration authority. A user who needs to access the thing, Then the thing sends an authentication request to the related registration authority. Registration authorities are distributed in different areas that cover all things. Registration authority requests ID from the user, after the user responds, Registration authority verifies the user‘s information and then sends an ID verification request to Head Registration authority to complete the verification process. Then Head Registration Authority asks the user for identification using any authentication techniques such as User ID and Password; the head registration authority responds if ID is authenticated or not. Once the ID received to Registration authority, it responds to the Thing about the user ID and grants access [2].

The registration authority verifies the content and identity of the Thing and reviews the content to determine if the information describes the use. Only authenticated user among the IoT can access the network to get the requested service [2], [10].

In general, IoT existing authentication schemas support IoT device local authentication, and IoT to gateway authentication, and IoT to cloud authentication. Authentication and security services with MEC providers is not addressed well in such solutions.

V. Conclusions

Mobile Edge Computing (MEC) providers can be used to improve the efficiency and resources utilization for IoT applications. Using third-party MEC providers as a middle layer between IoT devices and application centers introduces a set of security challenges for IoT applications. In this paper, we proposed a solution that addresses authentication, encryption, and integrity concerns that might arise from using third party MEC providers without compromising the efficiency obtained using MEC providers. In addition, the solution addresses mobility issues of IoT devices attached to different MEC providers. The solution addresses the mobility for one types of IoT scenario, and IoT client that talks to an IoT data center.

References

[1] V. Petrov, S. Edelev, M. Komar, and Y. Koucheryavy, “Towards the era of wireless keys: How the IoT can change authentication paradigm,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE, mar 2014. [Online]. Available: https://doi.org/10.1109/wf-iot.2014.6803116

[2] J. Liu, Y. Xiao, and C. P. Chen, “Authentication and access control in the internet of things,” in *2012 32nd International Conference on Distributed Computing Systems Workshops*. IEEE, jun 2012. [Online]. Available: https://doi.org/10.1109/icdcsw.2012.23

[3] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, “A survey on the edge computing for the internet of things,” *IEEE Access*, vol. 6, pp. 6900–6919, 2018. [Online]. Available: https://doi.org/10.1109/access.2017.2778504

[4] K. Apampa, G. Wills, and D. Argles, “Towards security goals in summative e-assessment security,” in *2009 International Conference for Internet Technology and Secured Transactions, (ICITST)*. IEEE, nov 2009. [Online]. Available: https://doi.org/10.1109/icitst.2009.5402505

[5] O. O. Bamasag and K. Youcef-Toumi, “Towards continuous authentication in internet of things based on secret sharing scheme,” in *Proceedings of the WESS 15: Workshop on Embedded Systems Security*. ACM Press, 2015. [Online]. Available: https://doi.org/10.1145/2818362.2818363

[6] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Cell phone-based biometric identification,” in *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, sep 2010.

[7] K. Habib, A. Torjusen, and W. Leister, “A novel authentication framework based on biometric and radio fingerprinting for the iot in ehealth,” 2014.

[8] T. Borgohain, A. Borgohain, U. Kumar, and S. Sanyal, “Authentication systems in internet of things,” *CoRR*, vol. abs/1502.00870, 2015. [Online]. Available: http://arxiv.org/abs/1502.00870

[9] C. Schmitt, T. Kothmayr, W. Hu, and B. Stiller, “Two-way authentication for the internet-of-things,” in *Studies in Big Data*. Springer International Publishing, 2017, pp. 27–56.

[10] B. Ndibanje, H.-J. Lee, and S.-G. Lee, “Security analysis and improvements of authentication and access control in the internet of things,” *Sensors*, vol. 14, no. 12, pp. 14 786–14 805, aug 2014. [Online]. Available: https://doi.org/10.3390/s140814786