Titre: Title:	Securing Mobile Payments in Connected and Not Connected IoT Environments
Auteur: Author:	Jean Gerald Vincent Etibou
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Etibou, J. G. V. (2025). Securing Mobile Payments in Connected and Not Connected IoT Environments [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/71643/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/71643/
Directeurs de recherche: Advisors:	Samuel Pierre
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Securing Mobile Payments in Connected and Not Connected IoT Environments

JEAN GERALD VINCENT ETIBOU

Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosphiae Doctor*

Génie informatique

Décembre 2025

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée:

Securing Mobile Payments in Connected and Not Connected IoT Environments

présentée par **Jean Gerald Vincent ETIBOU**

en vue de l'obtention du diplôme de *Philosophiæ Doctor*

a été dûment acceptée par le jury d'examen constitué de :

Alejandro QUINTERO, président
Samuel PIERRE, directeur de recherche et membre
Ranwa AL MALLAH, membre
Roch GLITHO, membre externe

DEDICATION

To my parents, my brothers and sisters, my son, and my partner.

ACKNOWLEDGEMENTS

I would like to thank my friend and long-time brother Panh COULIBALY for all his advice and assistance from India, Côte d'Ivoire, and Canada.

I would like to thank the AGOH family, Maman Virginie, Papa Jean François, Anna and David, Hervé and Yann for welcoming me into their home.

I would like to thank my cousin Hermann KAKOU for his advice, recommendations, and assistance from France and Luxembourg.

I would like to thank LARIM, Prof. Pierre, Dr. Eng. Franjieh, and my fellow students for your collaboration in the laboratory.

I would like to thank FLEX for their industrial expertise.

I would like to thank Polytechnique and GIGL for their academic support.

RÉSUMÉ

Le sujet de notre thèse s'intitule comme suit : sécurisation des paiements mobiles dans les environnements connectés et non-connectés.

Les objectives visés par ce travail est la sécurisation des paiements qui concernent tous les mécanismes de sécurisation des terminaux mobiles de paiements, des logiciels de paiements, des protocoles de réseaux supportant ses paiements dans les environnements connectés et non connectés.

Plus précisément, les environnements connectés qu'on a aujourd'hui, ce sont les environnements des réseaux et qui sont les réseaux internet des objets et aussi les réseaux qui premièrement connectent les terminaux de paiement des utilisateurs et qui sont en transition avec les réseaux des fournisseurs d'accès à internet qu'on trouve chez les clients. D'abord, ces réseaux ont été principalement chez les clients qui sont ceux qui reçoivent d'abord les connexions réseau locaux de petite taille supportant des équipements de paiement mobile. Ensuite, ces réseaux sont connectés à des réseaux plus grands que sont les réseaux des fournisseurs d'accès à internet. Les terminaux de paiement des clients sont connectés à ces grands réseaux que sont les réseaux de transport qui permettent de transporter l'information, qui sont aussi les réseaux apparemment connectés, mais aussi les réseaux qu'on ne voit pas connectés directement par les utilisateurs car l'étendue du réseau à grande échelle dans les réseaux des internet des objets.

L'hypothèse faite de la sécurisation des paiements mobiles revient à émettre une sous hypothèse qui est la sécurisation des réseaux qui sont connectés et non connectés. Cette approche reviendra à simplement sécuriser les réseaux que les clients s'aperçoivent voir, à partir de leur réseau local immédiat. Et aussi les réseaux que les clients ne voient pas et qui ne sont pas directement connectés à leurs terminaux de paiement. Cependant ces terminaux utilisateurs font partie des réseaux du service, de fournisseur d'accès et réseau ou internet.

La méthodologie appliquée dans ce travail sera axée sur la simulation d'un environnement hautement hétérogène ou un environnement internet des objets représentant une architecture compatible avec les paiements mobiles. Cette architecture sera proposée et implémenter suivant les techniques des technologies telles que les réseaux définis par logiciels (*SDN*) et l'informatique de proximité (*Edge computing*). Ensuite, dans cette architecture, il va falloir être en mesure

d'identifier les capacités des équipements et terminaux de paiement qui pourront être connectés pour voir effectivement leur niveau de connexion. Et ensuite fait en sorte que ce niveau de connexion qui permette de les regrouper par catégories et de permettre d'utiliser les potentialités de connexion que sont la localisation et le GPS afin de pouvoir être utilisé pour faire de la protection de ces paiements. Ensuite, l'autre partie de la méthodologie servira à pouvoir mettre en place ne algorithme qui pourra plus ou moins vérifier que tous les éléments que contribueraient à sécuriser la connexion, soit protéger de manière modulaire. C'est-à-dire que chaque niveau du système de protection sera pris en compte lors de la protection de la couche terminaux à la couche application. Les expériences de simulation et les analyses et les résultats obtenus ont pu prouver que la protection modulaire donne plus d'avantages de protection et que les résultats se feront testés et valider à partir d'une carte de test Arduino.

Ainsi donc, l'article numéro un au chapitre 3 de note thèse traitera d'un *framework* de sécurité inter environnement qui permettra de faire en sorte que l'architecture du *framework* de sécurité inter environnement proposée pourra être sécurisé à partir des technologies comme le réseau définit par logiciel sera capable de capturer les attaques de type dénis et de services. Mais celui-ci aussi permettra d'améliorer d'autre part la qualité de service et de faire en sorte que tous les équipements des utilisateurs qui communiquent sur le réseau de l'internet des objets puissent être protégés.

De même, l'article numéro deux, qui lui aussi va permettre de proposer une solution de sécurisation et d'identification des équipements utilisateurs dans le cas ces équipements auront des ressources de calcul limitées. Cette protection finale implémentée pour la sécurisation des paiements sera présentée sous forme d'une unité sécuritaire appelée un *modular security engine* (SE). Une protection de trois modules combinés que sont le DSM, NSM et SFB. L'implémentation en une unité permettra de faire évaluation de la protection par le même moyen d'une carte de test Arduino MKR 1400 compatible GSM, géolocalisation et internet des objets. Dans ce cas, notre article se heurtera à la limitation de calculs de données de grande échelle provenant d'équipements dotés de fonctions d'intelligence artificielle dans un environnent interconnecté des internet des objets avec des applications et terminaux de paiement. Nos expérimentations nous ont permis d'obtenir des résultats de chaque module composant du SEI. En observation conclusive, nous pourront déduire que la protection modulaire peut être effectuée grâce à l'interaction avec les utilisateurs du point de vue de la protection de des informations qui ont été demandées et reçues par les applications par l'échange de clés SSL et aussi par l'acceptation du message unique avec réponse aux clients. Cependant, nous noteront que les utilisateurs de services demandés tels que les transactions de paiement ont des performances variées dépendamment de la capacité de l'équipement Arduino qui a été testé.

Finalement, l'article numéro trois va offrir une solution de continuité de la protection en cas de non-fonctionnement du *modular security engine* à l'instar du proposé *cyber physical security engine* (CPSE) en utilisant la réplication digitale et le transfert sécurisé de profile d'équipement *IoT* afin de valider les informations et les résultats obtenus à partir des 2 premiers articles. Cet article 3 va nous donner une orientation plus étendue pour l'implémentation complète de la *sécurisation des paiements mobiles dans les environnent connectés et non connectés*.

On dira que ce travail peut être recommandé dans le cas d'études de développement d'applications de classes moyennes et de l'application des classes élevées sur la vérification de des informations de localisation. On peut aussi appliquer ce travail sur l'utilisation des capteurs récepteurs compatibles avec l'internet des objets. On peut aussi faire une autre application dans le cas où on veut procéder à la continuité de service de payment dans un lieu, une position géographique initiale ou l'un des capteurs se retrouveraient soudainement dans un environnement de faible couverture GPS dû à la complexité de mobilité des utilisateurs. Cependant nous recommandons que les utilisateurs de services en question soient enregistrés et que les équipements en question aient des identifiants universels qui peuvent être transmis et récupérés sur le réseau. De ce fait la continuité de service à travers la réplication digitale servira à faire la protection des communications transmises dans l'optique de pouvoir faire une protection complète si ces équipements qui participeraient à un échange de données de paiement. En addition, ce travail doit pouvoir être recommandé dans des cas ou les équipements géolocalisables sont participatifs dans des situations de vérification d'identité d'utilisateurs et de non-répudiation et cela me permettra d'améliorer les services de paiement globalement, à grande échelle.

ABSTRACT

The subject of our thesis is titled as follows: securing mobile payments in connected and non-connected environments. The objectives pursued by this work are the security of payments that concern all the mechanisms for securing mobile payment terminals, payment software, and network protocols supporting these payments in both connected and non-connected environments. More specifically, the connected environments we have today are network environments, which include the Internet of Things networks and the networks that primarily connect user payment terminals, transitioning with the networks of Internet service providers found at client locations. Initially, these networks were mainly at the clients, who are those that first receive small-scale local network connections supporting mobile payment equipment. Then, these networks are connected to larger networks which are the networks of internet service providers. The payment terminals of customers are connected to these large networks which are the transport networks that allow for the transportation of information, which are also the apparently connected networks, but also the networks that are not directly visible to users due to the vast scale of networks on the internet of things. The assumption made about securing mobile payments comes down to making a sub-assumption which is the security of both connected and non-connected networks. This approach will simply involve securing the networks that customers perceive, starting from their immediate local network. It will also involve securing the networks that customers do not see and that are not directly connected to their payment terminals. However, these user terminals are part of the service networks, access provider networks, and the internet.

The methodology applied in this work will focus on simulating a highly heterogeneous environment or an Internet of Things environment representing an architecture compatible with mobile payments. This architecture will be proposed and implemented following the techniques of technologies such as software-defined networks (SDN) and edge computing. Next, in this architecture, it will be necessary to identify the capabilities of the equipment and payment terminals that can be connected to effectively assess their connectivity level. Then, we will ensure that this connectivity level allows for categorization and utilizes connectivity potentials such as location and GPS to facilitate the protection of these payments. Next, the other part of the methodology will serve to implement an algorithm that will be able to somewhat verify that all elements contributing to securing the connection are protected in a modular way. This means that each level of the

protection system will be considered during the protection from the terminal layer to the application layer. The simulation experiments, analyses, and results obtained have shown that modular protection offers more protection advantages and that the results will be tested and validated using an Arduino testing board.

Thus, article number one in chapter 3 of our thesis will address an inter-environment security framework that will ensure that the architecture of the proposed inter-environment security framework can be secured using technologies such as software-defined networking, which will be capable of capturing denial-of-service attacks. But this will also help improve the quality of service and ensure that all user devices communicating over the Internet of Things network can be protected.

Likewise, article number two will also provide a solution for securing and identifying user devices in case these devices have limited computational resources. This final protection implemented to secure payments will be presented in the form of a security unit called a modular security engine (SE). A protection made up of three combined modules, namely DSM, NSM, and SFB. Implementing it as a single unit will allow for assessing the protection using the same means as an Arduino MKR 1400 test card compatible with GSM, geolocation, and Internet of Things. In this case, our article will encounter the limitation of large-scale data calculations from devices equipped with artificial intelligence functions in an interconnected Internet of Things environment with payment applications and terminals. Our experiments have allowed us to obtain results from each module that makes up the SE. In conclusion, we can deduce that modular protection can be achieved through interaction with users from the perspective of protecting information that has been requested and received by the applications through SSL key exchange and through the acceptance of the unique message with responses to clients. However, we will note that users of the requested services, such as payment transactions, have varying performances depending on the capacity of the tested Arduino equipment.

Finally, article number three will offer a continuity solution for protection in case the modular security engine fails, like the proposed cyber physical security engine (CPSE), using digital replication and secure transfer of IoT device profiles to validate the information and results obtained from the first two articles. This article will provide us with broader guidance for the complete implementation of mobile payment security in both connected and disconnected environments.

It can be said that this thesis work is recommended for the case of studies on the development of mid-range applications and high-class applications on the verification of location information. This work can also be applied to the use of receiver sensors compatible with the Internet of Things. We can also make another application in the case where we want to ensure the continuity of payment service in a location, an initial geographical position, or where one of the sensors suddenly finds itself in an environment of weak GPS coverage due to the complexity of user mobility. However, we recommend that the users of these services be registered and that the relevant equipment has universal identifiers that can be transmitted and retrieved over the network. Thus, the continuity of service through digital replication will help protect the transmitted communications with a view to providing comprehensive protection if these devices were to participate in a payment data exchange. In addition, this thesis work result could be recommended in cases where geolocatable equipment is participative in user identity verification situations and non-repudiation, and this will allow me to improve payment services overall, at a large scale.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

2G	2nd Generation Mobile Networks
5G	5th Generation Mobile Networks
API	Application Programming Interface
AIS	Automatic Identification System
IoV	Internet of Vehicle
OTP	One Time Password
POS	Point Of Sale
RNC	Radio Network Controller
EMV	Constraint Satisfaction Problem
PCI	Payment Compliance Industry
MEC	Mobile Edge Computing
SDR	Software Defined Radio
SDN	Software Defined Networking
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IaaS	Infrastructure as a Service
NFC	Near Field Communication
RSU	Roadside Unit
THE	Operating System
UAV	Unmanned Aerial Vehicle
PaaS	Platform as a Service
PDA	Personal Digital Assistant
DTN	Delay Tolerance Network
QoS	Quality of Service
VR	Validity Rate
SaaS	Software as a Service
TCR	Transaction Completion Rate

IoT	Internet of Things
To	Tera octet
NS	Network Simulator
EN	Edge Server
SE	Security Engine
CPSE	Cyber Physical Security Engine

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Nowadays, the mobile telephony field has undergone very big transformations from the point of view of its traditional services such as voice, data and messaging [1].

These transformations occur in the sense that these traditional services are used to offer other services on top of what are called value-added services [2].

These value-added services are in several categories, the most popular of which is mobile payment.

Indeed, mobile payment allows telecom operators to expand their portfolio of services by addressing different types of users who no longer only care about making a voice call, or sending a text message sms or mms, but about being able to send or receive money electronically [3].

As the user is at the center of this transactional activity, it must comply with user data protection standards throughout the process.

It is noticeable that users of mobile payment services have more mobile communication terminals that would potentially be capable of sending and receiving payment transactions. As a result, all these pieces of equipment need to communicate constantly. However, a suitable platform must be set up for this purpose [4].

Mobile payment is of increasing interest to researchers, so that both mobile and telecommunications payment environments are becoming increasingly heterogeneous. This is translated into the Internet of Things IoT [5].

This heterogeneity opens many areas of research covering payments with connected objects "Internet of Things" (IoT) and blockchain [6].

However, the problems inherent in payment, such as connection loss, authentication failure, and lack of protection of user data, demonstrate that there are contributions to be made to improve payments in highly heterogeneous (i.e., IoT) environments [7].

It is also important to mention that the management of telecom services is influenced by a new paradigm of software-defined networking - SDN, which allows for the deployment of a scalable architecture that allows flexibility in the management and control of services and endpoints [8][9][10].

In addition to being flexible, telecom architecture should allow terminals with low computing power and hardware resources to be less energy-intensive and more efficient. This is the ideal scenario for mobile payments and leads to the adaptation of an architecture supporting another paradigm that is *mobile edge computing* – MEC [10][11].

It should be noted that the payments industry is full of very few essential players who offer payment services at very competitive costs while guaranteeing a quality of service intimately linked to user requirements [11][12].

As a result, we focus this thesis on the creation of a new mobile payment environment to support Internet of Things users' terminals used to make offline and online transactions, considering the advantages offered by *software-defined networking* and mobile *edge computing*.

This document is organized as follows. In Section 1.1, we first introduce the different basic concepts related specifically to mobile payment and which will be essential to the understanding of mobile payment security in this thesis. In Section 1.2, the elements of the problem are presented. Then, in section 1.3, the research objectives are defined, followed by the outline of the thesis in section 1.4.

1.1Definitions and Basic Concepts

In this chapter, we propose to define the main concepts related to *edge-computing-assisted* for mobile payment. To do this, we start by presenting mobile payment and its main features, the various advantages justifying such interest in users, as well as the limitations and research challenges related to it. Next, we try to explain the key terms that allow us to better understand *edge-assisted computing* for mobile payment in *IoT environments*.

1.1.1 Definitions

Mobile Payment

The definition proposed by the literature defines mobile payment as a payment service made from a mobile device instead of a chip bank card [13].

Mobile Edge Computing (MEC)

Edge computing or edge computing, a key technology for improving computing resources, is being introduced into blockchain-enabled IoT applications, for example. These applications can help IoT devices offload compute-intensive tasks to edge servers (ESs) or edge servers [14][15]. In the edge-assisted IoT-based blockchain, the trading of computing resources inevitably occurs between edge servers (ES) and IoT devices.

Internet of Things (IoT)

Connected objects or IoT is a network of electronic devices with the ability to collect and transfer data from the environment and interact with remote computers over the Internet [16]. An IoT device is anything that could transfer data over the internet to communicate with other connected devices, which means they are assigned IP addresses. These devices can be anything, such as cell phones or smart lights.

IoT devices cover a wide variety of applications worldwide, such as healthcare, transposition, agriculture, industry, market, smart home, smart school, smart city, and vehicles, etc. [17][18].

1.2Basic Concepts

1.2.1 Mobile Payment

The electronic payment industry is constantly diversifying and improving, to meet the needs of users. The adoption of mobile payment on the market has been largely encouraged by the development of technology compatible with mobile retail terminals, as well as mobile *broadband* or mobile data [19].

Mobile payment refers to the use of a terminal or a mobile application, connected to a payment network offered by the payment service provider, for the purpose of issuing or receiving a financial transaction. Financial transactions that once took place within a financial institution only can now be carried out from anywhere in the world. Mobile payment users can now access online services, without having to go to their bank. This allows them to save a lot of money in terms of travel costs and greater freedom of movement and productivity. Although it offers great flexibility and convenience for users, mobile payment has yet to be adopted in much of the rest of the world, especially in countries with low internet connectivity rates [20].

Mobile payment is essentially made up of six components [21]:

- Users (user device): Sends and receives transactions (payment request);
- Vendor (merchant device): Accepts and transmits payment information;
- MP Gateway: Represents a logical interface routing the transaction;
- Processor (MP Processor): Represents an interface for routing the payment request;
- Acquirer (MP Processor): Represents an interface for transmitting the payment request;
- Card issuer (CC, DC, Bank cards): Represents a financial institution checking creditworthiness;
- Payment Network: Represents a heterogeneous interface for the transit of the payment request.

Mobile payment, as shown in Figure 1.1, generally works in one of two ways [22]:

Online: Mobile payments made from user terminals that always require connection to a data network at the time of initiating the transaction.

Offline: Mobile payments made from or on terminals not connected to a data network at the time of initiating the payment transaction.

Contactless: Mobile payments made with user devices with contactless technology such as Near Field Communication (NFC).

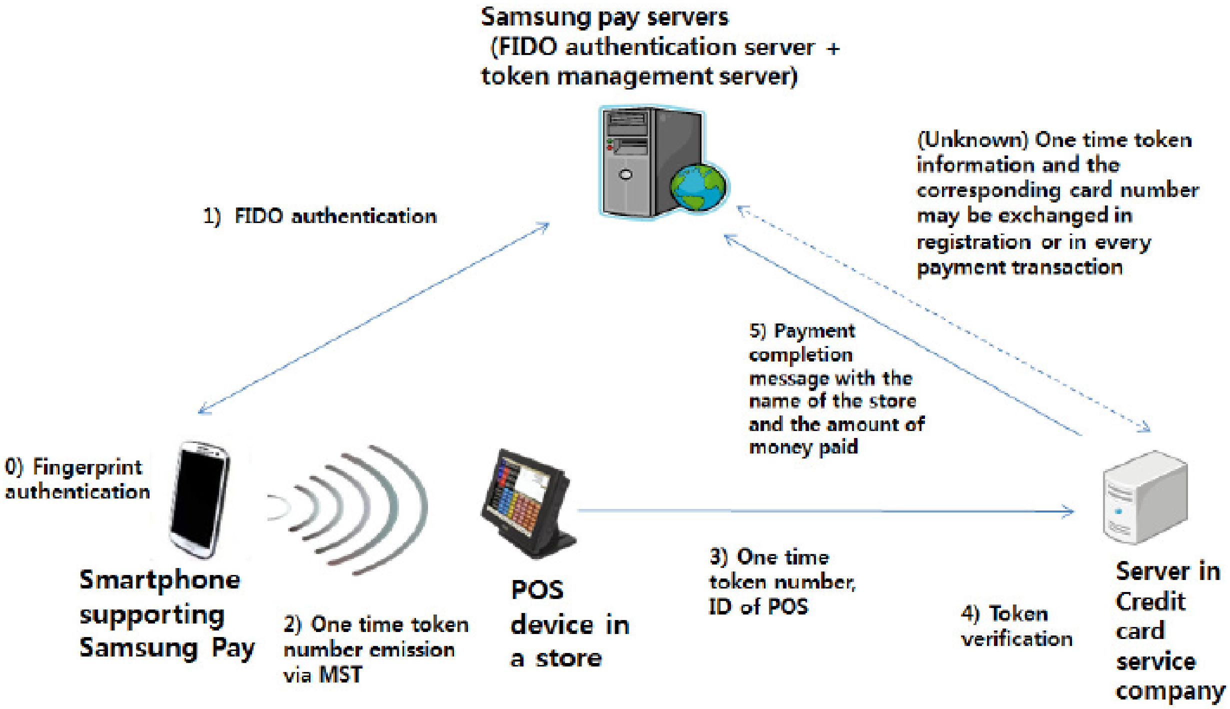

Samsung pay servers
(FIDO authentication server + token management server)
(Unknown) One time token information and the corresponding card number may be exchanged in registration or in every payment transaction
1) FIDO authentication
5) Payment completion message with the name of the store and the amount of money paid
0) Fingerprint authentication
Smartphone supporting Samsung Pay
2) One time token number emission via MST
POS device in a store
3) One time token number, ID of POS
4) Token verification
Server in Credit card service company

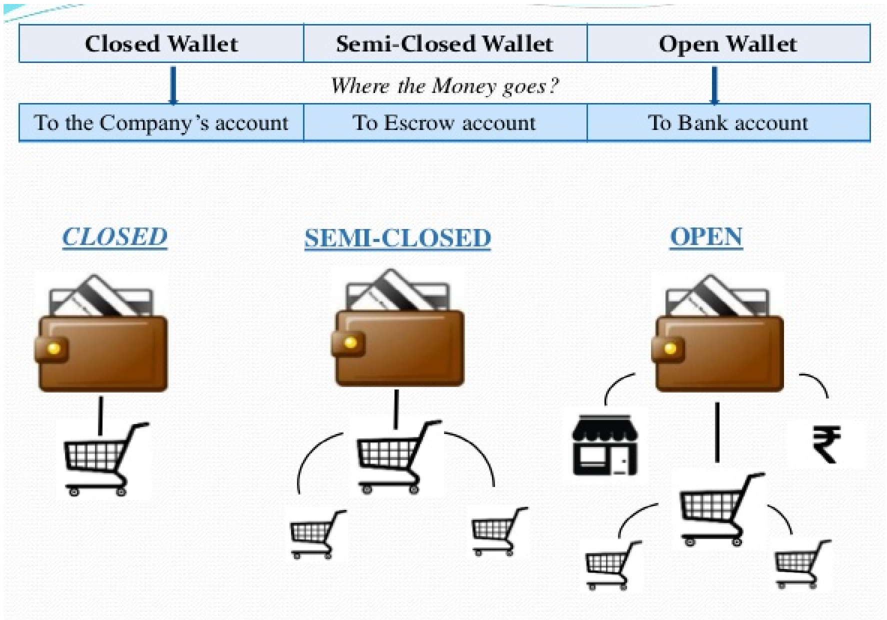
Figure 0.1 Service Operating Models [22]

Depending on its architecture and implementation, mobile payment can be deployed in four types [23]:

- Private (proprietary) payment: This payment model allows a business to provide multi-type payments to a consumer for the purchase of goods and services, as shown in Figure 1.2.
- Closed payment: This payment model is made by a company to a consumer for the purchase of goods and services exclusively from the company, such as Amazon, Best buy, Walmart, and others.
- Semi-closed payment (semi-closed wallet): In this payment model, the purchase of goods and services is made on clearly identified sites or merchant establishments, such as PayPal and

Paytm. Prepaid services are used. This payment does not allow cash withdrawals or redemptions.

- Open payment (open wallet): This model, however, allows it to be used for the purchase of goods and services, for funds intended for withdrawing money from ATMs, such as Mastercard, Visa and Amex.



Closed Wallet	Semi-Closed Wallet	Open Wallet
	Where the Money goes?	
To the Company's account	To Escrow account	To Bank account

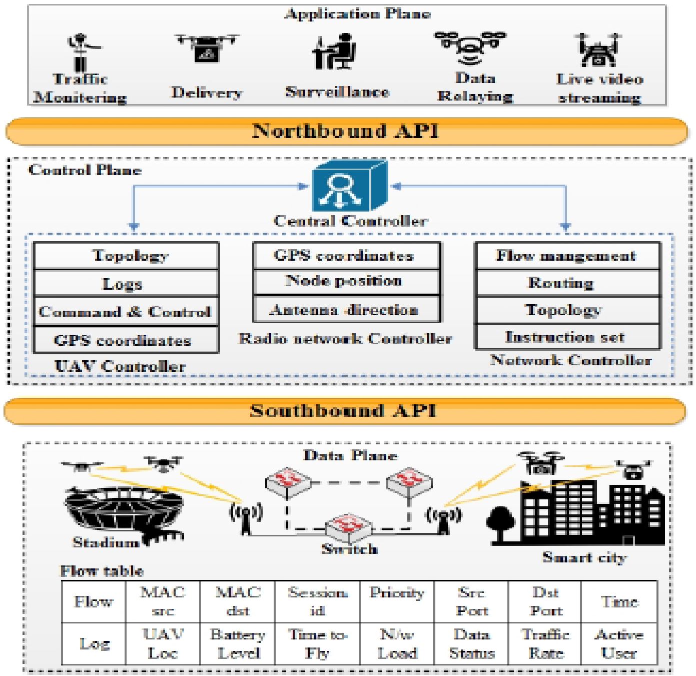
Figure 0.2 Payment Type [23]

1.2.2 Software Defined Networking pour Paiement Mobile (SDN)

Nowadays, communication networks are becoming increasingly manageable through management software that is used to control the elements that make them up. It is in this sense that the concept of Software Defined networking (SDN) plays a very essential role in the representation of the network in three major dimensions. Indeed, SDN was introduced as a concept in which the separation of the control plane and the data plane played a very significant role in the architecture of communication networks [24]. Generally speaking, there are three different plans:

- Application Plan
- Control plane
- Data plane

SDN was introduced in the last decade, in case the separation of data and control planes play a big role in the network architecture. In SDN, the control plane acts as the logical centralized brain and is maintained by the OpenFlow protocol [24]. Today, it is considered the most reliable and efficient mechanism for error-free operations and drone deployment, as shown in Figure 1.3.



Flow	MAC src	MAC dst	Session id	Priority	Src Port	Dst Port	Time
Log	UAV Loc	Battery Level	Time to Fly	N/w Load	Data Status	Traffic Rate	Active User

Figure 0.3 Architecture of the SDN System Model [24]

1.2.3 Mobile Edge Computing for Mobile Payment (MEC)

MEC, originally introduced by IBM, aimed at optimizing the existing mobile infrastructure service, and minimizing the average delay of general traffic in the LTE downlink [10]. Due to its characteristics of low latency, on-premises *presence*, and *location awareness*, MEC is being introduced into architecture to improve responsiveness for IoT and Internet of Vehicle IoV applications. The MEC server can be deployed on 5G for example, the situation close to the end user at the edge of the IoT network, and of each application running on this network. In addition, MEC makes it easy to migrate cloud services and virtual machines, view data, and synchronize it within a uniform framework of the management infrastructure.

The MEC framework, as shown in Figure 1.4, consists of the MEC hosting infrastructure, the MEC application platform, and the application layer. The MEC hosting infrastructure consists of a hardware virtualization layer and a set of hardware resources, especially communication devices. The MEC application layer of the platform includes the infrastructure as a service and a set of middleware (i.e., service management, communication components, cognitive function of radio information network and traffic offloading). Applications such as mobile payment applications are deployed using virtualization of independent machines.

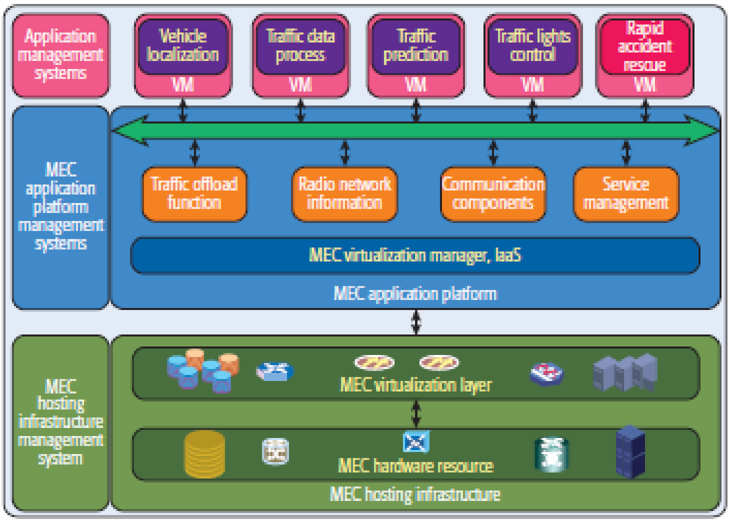
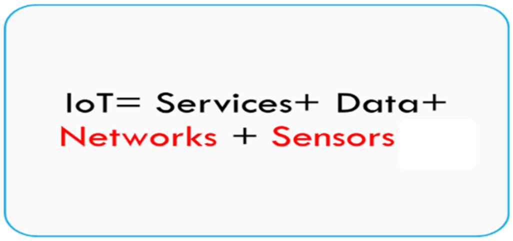


Figure 0.4 Framework MEC [10]

1.2.4 IoT Environment for Mobile Payment

Connected objects make up the system of physical objects in the world that connect to the Internet via a sensor. The IoT components have been summarized as a combination equation of sensors, physical objects, controller, and actuators as shown in Figure 1.5 below.



IoT= Services+ Data+ Networks + Sensors

Figure 0.5 Iot and Smart City concepts [26]

The term "Internet of Things (IoT)" acts as an umbrella keyword that covers all the different applications of the Internet, such as extending the Internet, the web as a physical domain, deploying extended embedded distributed devices, sending, and actuation capabilities. This means that mobile payments will also be part of the daily life of IoT, especially with the adoption of Broadband IoT [25] in connected environments.

The ubiquitous trend related to computer technology of the latest technology in the form of connected objects. In IoT, things allow the connection to the Internet to work anytime and anywhere using different technologies. The four main categories of key technology made available by IoT include:

- To tag the things RFID technology uses
- to detect things that sensor technology is using
- to think about the things that smart technology uses
- to reduce the things that nanotechnology uses

IoT is very famous for the biggest improvement in the latest field of ubiquitous computing, such as Wireless Sensor Network (WSN) and Machine-to-Machine (M2M)-based communication.

IoT is a fusion of heterogeneous networks, including chip technology which is increasingly expanding due to the rapid growth of internet applications such as logistics, agriculture, smart community, intelligent transposition, control and tracking systems. Future projections suggest that IoT objects will be semi-intelligent and an important part of human social life.

1.3 Elements of the Problem

Mobile payment is essentially characterized by the flexibility it offers to make a financial transaction from almost anywhere the payment service network is available with or without a physical bank card. This ubiquity aspect of mobile payments fits well with IoT high-mobility environments.

However, the problem of securing mobile payments prevails. Moreover, considering the context of *the IoT* with the increasing number of connected terminals, the process of securing mobile payment transactions and mobile payment user data remains a relatively complex problem to consider and solve.

To do this, we will look at the issues that would cause the security vulnerability of mobile payments in *IoT environments*.

First, we question the problem of non-connectivity directly linked to the frequent partial or total interruption or break of the high connectivity or always connected *network links* that the IoT promises. The main challenge is to find a scientific and technological solution to allow reconnection with the infrastructure during a partial and total outage. Indeed, we would like to specify that we have no control over the choice of mobile payment providers or the configuration parameters of their system, such as the speed of transmission of links, the interconnection of the different nodes or the maximum capacity of the latter. However, it is a matter of trying to specify the normal conditions for making payments in connected and non-connected mode, as well as to develop mechanisms to maximize the use of the range of means offered to users. Moreover, it adds a degree of difficulty in the process of securing mobile payment.

Second, we looked at the challenge of localization in an *IoT* environment for mobile payment applications running on users' mobile devices. Indeed, the presence of several communication devices in the same environment poses the problem of densification in a communication network. There are two types of densifications: horizontal densification and vertical densification. Horizontal densification represents occupancy in a flat era without an increase in the number of communicating devices, whereas vertical densification would represent an occupancy of an era in elevation or height. It follows that known problems of densification in literature can help to find different methods of solving this kind of problem. However, real-time localization in a high-density environment makes solving this problem even more complex. We will be able to combine approaches by focusing on performance and data security.

In addition to the complexity of our work, it can be deduced from the above that mobile payment users could be confronted with the problem of the absence of payment tokens. By way of explanation, the *payment token* can be in the physical form, in this case a traditional plastic bank card, or in the dematerialized or digital form such as crypto-currency or mobile *wallet*. As a result, it is likely that a third-party user could forget or even misplace his bank card or even the latter could prove to be non-functional at the time the user decides to use it by presenting it at transaction terminals such as ATMs or POS points of sale. It is in the sense of avoiding these inconveniences that digital solutions such as cryptocurrency and e-wallets have been developed. However, these technologies are functionally dependent on connectivity to a data network. So, if there was a connection break, the user would not be able to use them. Thus, it would be very wise to find a better solution that would allow the payment service to continue even in cases of loss *of payment token* or non-connectivity. To this end, it is essential that the mobile payment application running on a terminal could retrieve, send and receive the location independently of the connectivity of the infrastructure, to send and receive financial transactions. The important thing is that the operation and performance of this payment guarantee the security of user data in all uses.

Another point of interest is the lack of user privacy due to the class of service that users interact with by submitting sensitive personal data through the apps' interaction interfaces.

Indeed, the lack of standards in the field of mobile payments and the monopoly of certain solutions belonging to manufacturers of mobile terminals or operating systems generate doubts about the practices used when processing transactions that would guarantee the maintenance of privacy. This can be explained by the fact that payment terminal manufacturers can dictate which technology their terminals support or not and decline the operation of the payment service with other technologies. Therefore, to better guarantee the secure processing of user data, it would be reasonable to produce a validation of the mobile payment security parameters according to the standards that the payment industry recommends. Then, it is a question of implementing mobile payment security and evaluating any form of persistence of geolocation data.

A second concern is the security expectations of mobile payment users. It is due to analyzing the different types of attacks perpetrated against mobile payment users with the aim of stripping them of their assets. Indeed, research on the security of mobile payment systems has reported risks such as user masquerade, malware, man-in-the-middle attacks, and traffic interception. In addition, the lack of two-factor authentication increases the chances of sending money to the wrong account or passing a password.[26] Financial institutions have responded to these threats by providing consumers and dealers with licensed service managers who approve and verify all mobile transactions [27]. This process is aimed at user customers, and third parties trust the system and ensure the security of all partners. The sensitive nature of transactions between banks and mobile payment service providers also requires that any conflicting issues be dealt with effectively to achieve the objectives set [28].

As such, this makes it clear that both attacks can be triggered against the client device or transaction protocol, as well as attacks aimed at customer-sensitive data. Depending on the capabilities and number of devices that attackers can access during the attack, an attacker taxonomy is first introduced as follows [21]:

- Collector: This is an external attacker capable of listening to and modifying messages exchanged between the client and the provider device;
- Malicious client: This is an insider attacker who can either physically open the client device to listen to sensitive information or inject malicious code into the client device, to modify its behavior;
- Malicious provider: This is an insider attacker who can either listen to information from the provider device or inject malicious code into it, to modify its behavior;
- Ubiquitous: This is an insider attacker with full access to all devices involved.

However, it is important to mention offline scenarios that are more difficult to protect. In these cases, customer data is kept in the *point of sale* (POS) terminal for much longer, which is more vulnerable to attackers.

Related location services in *IoT* environments are of the greatest challenges to the viability of mobile payments. This extends further with the densification of IoT devices that could potentially make payments from IoT broadband networks. Hence the importance of finding optimal mechanisms for the collaboration of resources that can be used to do protection by *location awareness.*

Looking in a little more detail at the origin of mobile payment transactions, the lack of localization of transactions would be to the detriment of user-initiated payment security measures.

Thus, in the light of those difficulties, the answers obtained for each of those problems, taken individually, constitute partial solutions which are combined, with the help of integration mechanisms, to produce the best solution.

However, although the decomposition into sub-problems seems to be less complex to implement, no implementation in the direction of our project has yet been implemented or integrated and even developed yet, not to mention ensuring the optimality of the final solution.

Therefore, to consider the various issues that are important to face in the process of securing mobile payment in *an IoT* environment, a multi-objective approach is strongly recommended to produce a payment model adapted to the current context.

Thus, the implementation of such a model is quite a complex task but would have the merit of providing excellent perspectives of solutions for connected cities.

The various elements of the problem set out above have therefore led us to ask ourselves the following main question:

- How to protect users' mobile payment transactions in IoT environments?

From this main question arises the following additional questions:

- What is the most relevant payment infrastructure that fits the requirement of today's mobile payment user?
- How do you enable payment transactions when there is both connectivity and non-connectivity?
- How to offer and maintain the privacy of the mobile payment user in the context of the user's environment in respect of offline and online cases?

- How to prevent unauthorized transactions by attackers, spoofers and fraudsters on payment platforms?
- How are mobile payments performance transactions improving significantly?

Thus, many questions led us to state the main research objectives presented in the next section.

1.4 Research Objectives

This thesis aims, mainly, to design a mobile payment security system based on a modular architecture in *IoT* environments in the perspective of optimal protection of user data, taking into account the interactions between the different parties participating in this transaction.

More specifically, this thesis aims to:

- Analyze existing non-mobile and mobile payment methods and architecture with a view to enriching the literature review;
- Design an architecture for mobile payment reinforced by a security framework offering IoT endpoint security and secure data routing in IoT environments;
- Design performance and device selection models contributing to security in connected and non-connected mobile payment in IoT environments;
- Implement a modular mobile payment security system with location verification factor and user authentication within the proposed architecture;
- Evaluate the performance and level of secure data exchange of endpoints within the proposed architecture.

1.5 Outline of the Thesis

In Chapter 2, we present a literature review on mobile payments in the context of mobile edge computing, Software Defined Networking (SDN), Internet of Things (IoT), and Smart City.

In Chapter 3, we will introduce the first part of this thesis denoted by our first article entitled: "SDN-Based Security Framework for Internet of Things' Payment".

In Chapter 4, we present the second part of this thesis denoted by our second article entitled: "IoT Devices Modular Security Approach Using Positioning Security Engine".

In chapter 5, we detail the third part of this thesis denoted by our third article entitled: "Modular Security Engine Cyber-Physical Resiliency Approach Using Digital Replication".

We conclude this thesis in Chapter 6 by reporting on the synthesis of the work, discussing the limitations of the proposed solution while suggesting the improvements that could be made in our future work.

CHAPTER 2 LITERATURE REVIEW

In this section, we will present the main work that has been carried out in the field of mobile payment in a Smart City or IoT environment. The process of securing user information aims to maintain privacy, non-repudiation, and authentication in a heterogeneous environment such as the Smart City or IoT.

Indeed, while some works aim to secure users' information online, others seek to secure offline payment terminals. However, some researchers also want to maximize user protection by exploring the environment in which users find themselves. As a result, for many of these researchers, it is becoming necessary to put in place mechanisms to secure the transactions of sedentary users or users in transit. However, other studies correlated with positioning and mobile devices would be of great contribution to the field of mobile payment, it is important to describe such an advantage for our thesis.

The work presented below in this literature review is grouped into concept categories as set out in section 1.1 of chapter 1 of this thesis.

2.1 Literature on the Security of Mobile Payment in a Connected Environment (*online)*

One of the main works of literature in the security of mobile payment is the privacy-preserving mobile payment system for public transport [13], whose objective is to protect the private data of any passenger in transit online. The methods used to achieve this goal are Pairing-Based Cryptography (PBC), Signature-Based Identification (GMP) and bi-linear combination and linear encryption cryptographic methods. These methods, combined in a system called the Transit Transfer Ticket (TTIK), as shown in Figure 2.1, make it possible to validate the passage of transit passengers through the electronic access gates to the departure areas. However, the limitations of the system would be explained by the fact that this system could only work if and only if there is an established connection to the network. So, it would be justified to ask the question of what would happen if there was no connection? Because of this, it would certainly be wise to consider a system that is tolerant of network disconnections.

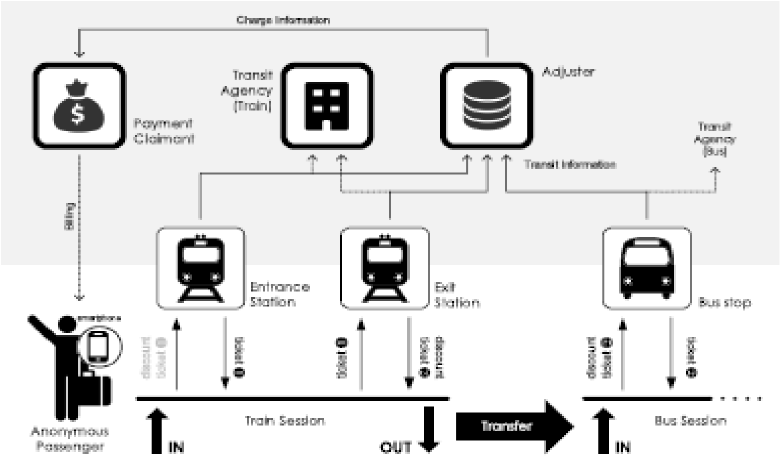


Figure 0.1 Online transit payment protection model [13].

To continue with the security of online payments, a work of the literature [29] has focused on the implementation of a robust mobile payment system with a transaction repository based on Blockchain smart contracts. The objective of this system is to secure online transactions for computationally constrained mobile devices, as shown in Figure 2.2. This would potentially be adequate for lightweight devices in an IoT environment. As far as the methods used in this work are concerned, it is a question of protecting payment mechanisms by using robust crypto-primitives, i.e. without the need for signature and certificate unlike previous work, more precisely the elliptic curve, bilinear matching, blockchain, smart contracts. However, there are limitations, which are among others the constraints of the blockchain mainly. Nevertheless, this implementation is highly recommended in the context of establishing the online connection between the sender of the transaction and the transaction processing resource.

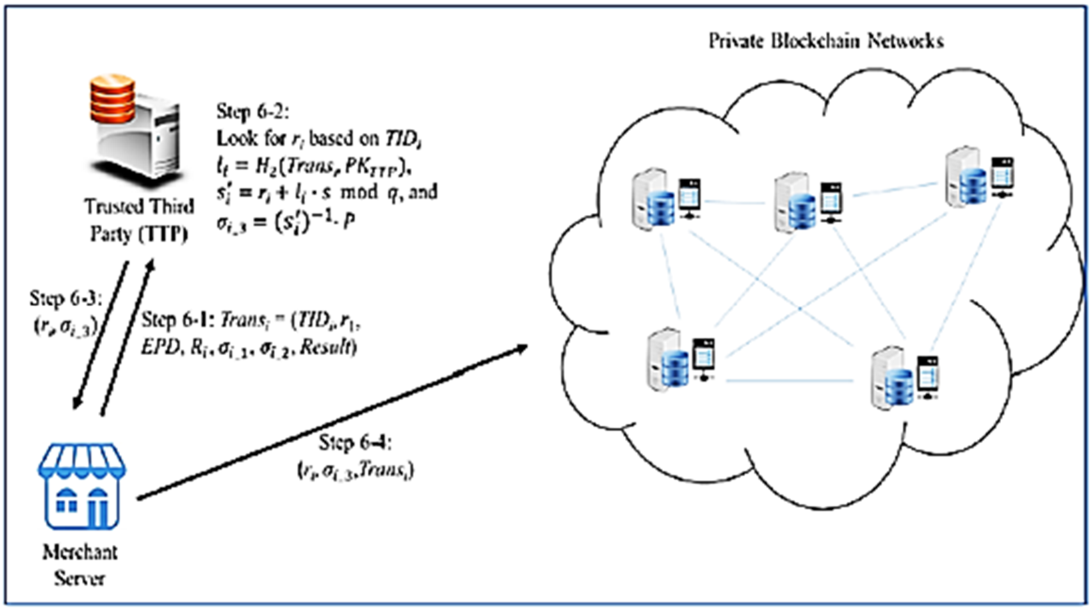

Private Blockchain Networks
Trusted Third Party (TTP)
Step 6-2:
Look for r_i based on TID_i
$l_i = H_2(Trans_i, PK_{TTP})$,
$s_i' = r_i + l_i \cdot s \mod q$, and
$\sigma_{i\_3} = (s_i')^{-1} \cdot P$
Step 6-3:
(r_i, σ_{i_3})
Step 6-1: $Trans_i = (TID_i, r_1, EPD, R_i, \sigma_{i\_1}, \sigma_{i\_2}, Result)$
Step 6-4:
$(r_i, \sigma_{i\_3}, Trans_i)$
Merchant Server

Figure 0.2 Robust model of online payment protection [29].

2.2 Literature on the Security of Mobile Payment in a Non-connected Environment (*offline*)

Market analysts have predicted that mobile payments will overtake the traditional market, providing greater convenience for consumers and new revenue streams for many businesses [30]. This scenario produces a shift from traditional credit card purchasing methods to new approaches such as mobile payments, while giving new market entrants new business opportunities.

It is widely shown that the payment equipment must be fully recent. This is justified by the fact that mobile payment technology is still in its early stages of evolution, but it is expected to increase soon, as demonstrated by the growing interest in cryptocurrencies.

Thus, the researchers looked at the methods and means of securing these payment terminals. It is in this same direction that the FRodo Fraud-Resistant Device for Offline Micro-Payments [21] is adopted, whose objective is to provide secure full payments while being resilient to all currently known point-of-sale or transactional (POS) breaches. The authors of this work use as methods secure hardware encryption based on strong non-clonable physical functions (PUFs) with:

- an element of identity;
- an element of electronic money.

To take it a step further, some authors have proposed a delay-tolerant payment system based on the Ethereum Blockchain [31] that aims to provide reliable digital payment services over unreliable networks in remote areas. To achieve this, the methods used are implemented in a digital payment architecture based on the Blockchain whose components are:

- with NFC-enabled payment gateway;
- ethereum blockchain;
- a smart contract-based manager.

In addition, the authors of the work on a Secure Payment System using Mobile Ad Hoc Network (MANET) for disaster areas [32], found a way to activate payment transactions in disaster areas as offline payment. This becomes interesting in that this work is part of the research area on emergency payment systems. The methods used by the latter include:

- an approval-based mechanism;
- a multi-level Mesh Link Establishment (MLE) approval mechanism,
- a Delay-Tolerant Networking (DTN);
- MANET without infrastructure;
- Location-based mutual monitoring.

All this work comes up against the problem of protecting transactions from the payment rental environment. This is because fraud attempts on payment systems can only be thwarted if and only if the consideration of the rental of transactions triggers a protection factor that must be validated by the user before the user's transaction can be completed.

2.3 Analysis of the Literature

As we have noticed, several authors have looked at the problem of securing mobile payments and particularly user data. However, a quick analysis leads us to believe that some avenues of research remain unexplored for payments in heterogeneous environments such as Smart City or IoT.

Indeed, with the promises that the Smart City or IoT will offer, the issue of security and performance is an important aspect to consider in the process of making mobile payments, especially in the context of increased heterogeneity. Indeed, some mobile payment applications will be able to be fully protected in some regions better than in others, or even in certain countries. As a result, mobile payment services are being implemented while taking into consideration the access to information policies and laws of some countries alongside the critical level of data hosted by servers located in these countries.

According to most of the literature literature, mobile payments run on different terminals or platforms depending on the user's choice of terminals. Thus, it suggests that the cost associated with the acquisition of mobile payment terminals on the client-user side varies due to aesthetic, technological and convenience preferences. In addition, the accessibility and availability of services should not be neglected. These can play a key role in the widespread adoption of mobile payments. As a result, it would be interesting to develop a more general mobile payment system that would consider these different particularities.

Finally, although several studies have been carried out in the field of mobile payment security, the increase in the number of mobile terminals in the world should not be neglected. However, today, despite efforts to minimize attacks and fraud, the result is that they are becoming increasingly diverse. However, after analysis, we notice that this factor is also increasingly known to users. However, they make a compromise between actively securing data by verification SMS OTP and convenience of securing by tools such as *TouchID*.

Thus, in the process of securing user data, it would be interesting to consider some specific options of the applications of the users' terminals.

CHAPTER 3 ARTICLE 1: SDN-BASED SECURITY FRAMEWORK FOR INTERNET OF THINGS' PAYMENT

Submitted on October 10th, 2025 to Journal of Cyber security and mobility by Jean Gerald Vincent Etibou and Samuel Pierre

Abstract— Internet of Things (IoT) ecosystem consists of heterogeneous inter-connected devices used by users in consumer services, such as mobile payment. Mobile payment has diverse and dynamic requirements for maintaining quality of services (QoS), maintaining network routing of protocols while ensuring end to end safety of user data over cross network environments with the IoT ecosystem. However, the coexistence within the same IoT ecosystem of different payment services running on different IoT devices becomes complex to integrate into a common security purpose framework because of the heterogeneity of IoT user devices, payment service protocols, limited inter devices communication trust, and exposure to vulnerabilities, threats and attacks such as Distributed Denial of Service (DDoS). In this paper, we designed an IoT architecture to implement a cross-environment security framework for IoT applications and devices based on Software Defined Network (SDN) backed by Rivest-Shamir-Adleman (RSA) to securely implement SDN for IoT. In the proposed cross IoT environment security framework, we first integrate an IoT device security model with location information factor verification to mitigate spoofing and impersonation attacks based on receiver radio signal (RSSI) and radio signal propagation. Secondly, we propose a trusted scheme for cross-environment IoT networks with data routing to provide reliable data routing and data confidentiality. The simulation results demonstrate improved security performance, increased network throughput, less package loss, low latency rate, and minimal computational overhead in an SDN-based test bed IoT ecosystem.

Index Terms— computational overhead, cross-environment, data routing, Internet of Things, mobile payment, security framework, Software Defined Network.

3.1 Introduction

The second millennium has witnessed an explosion of Internet of Things (IoT) connected devices aggregating an increasing amount of user and device data [33]. Access point, smart grids hubs , virtual healthcare check points, connected cars and more are among IoT connected devices enabled with characteristics such as heterogeneity, automation, intelligence, mobility and more, which bring challenges in designing a cross-environment framework aiming at a common purpose framework to solve security issues in IoT ecosystem such as limitation of device authentication and secure user and device data routing [34].

IoT ecosystem authentication and routing issues require robust authentication and trusted routing access control to network elements, user applications and resources [35].

Recently, security has been a key requirement to succeed in framework integration in every network, such as traditional networks (i.e., wired, wireless, ad-hoc, mesh) and modern networks (IoT) [36].

In modern network architecture, a purpose framework in security will work across two levels: the network level with data routing and the access level with authentication. Thus, our proposed framework should integrate a technology that provides the ability to differentiate the data plane from the control plan while allowing the user to take control of the network elements management, unlike traditional network [37].

In this modern network era, there is the emergence of the paradigm Software Defined Network (SDN). SDN is capable to split the data plane from the control plane. This decoupling of control and data plays a significant role in large-scale, high-speed computing systems. SDN deploys OpenFlow as a standard communication protocol used between the controller in the network plane and forwarders in the data plane. The overall SDN architecture includes OpenFlow switches, controllers, and data forwarders with flow entries [38].

In this paper, we will focus on payment systems including payments applications as one-use case among many existing sub-systems in the IoT ecosystem environment.

Our first concern with many IoT applications such as payment applications on IoT devices or terminals is that they are easily accessible to any adversary. Thus, they are subjected to physical and side-channel attacks. Therefore, an attacker can get remote access to legitimate IoT nodes and network resources via the exposed terminal and launch a spoofing attack by stealing the secret keys stored in the IoT devices or terminals. To solve this issue, we use location information factor authentication by verifying the IoT location or terminal location to mitigate attacks impersonation. The technique being used is the wireless channel characteristic in the form of the received signal power indicator (RSSI) of the radio signal of the IoT device or terminal. This technique will help to lock the location of the IoT device or terminal even in an advert that they are been cloned by the attacker or moved from their original designated location.

Our second concern with multiple IoT nodes or devices communicating statically or dynamically is that IoT nodes are unlike traditional devices, e.g., laptops or cellphones. They are physical objects such as cars, watches, and others. During communication, they are forwarding data information using multi-hop routes, which selection needs to be optimal and secure. The solution to the issues on the selection of the optimal route will bring a significant improvement in IoT-based applications such as mobile payment, especially in decreasing connectivity failure with increased network adaptability. Therefore, in this paper, we construct a reliable routing forwarding channel among IoT devices that forward data toward the IoT gateway using an IoT service router.

The novelty of this research work lies in designing an IoT cross environment security framework combining SDN and RSA for securing applications' data, such as payment data, as well as IoT devices from DDoS attacks within an IoT-based payment ecosystem.

The motivation behind the design of the cross-environment security framework lies in the fact that the IoT ecosystem interconnects many wireless nodes, statics or mobiles using wireless links capturing information that needs to be forwarded to an intermediate chain of IoT devices or network entities towards end-point users or network entities without failure. Therefore, an architectural design approach will present the imbrication of combined multiple frameworks, models and schemes for a better representation and simulation of the proposed solution will mitigate the issues of user authentication and secure user data routing, both prone to DoS attacks.

The main contributions of this paper are as follows:

1 *We design an IoT cross environment architecture using SDN and mobile edge computing (MEC). MEC is selected to maintain the delivery to all categories of participating IoT*

devices of the framework whereas SDN decouples the IoT cross environment in device, data and application layers at a large-scale [40].

2 *We implement a cross IoT environment security framework for securing payment as well as IoT devices through a robust authentication scheme within the detectable broadcast signal of user's IoT devices using RSSI wireless signal measurement technique to match the location of IoT user device or IoT end of service terminal [41].*

3 *We enhanced the proposed cross IoT environment security framework by including a trusted data routing scheme based on RSA featuring detection and prevention of attacks from malicious IoT sources devices, destination IoT devices, network controllers and forwarders IoT entities from participating in user data routing and forwarding [42].*

4 *We evaluate the proposed framework by simulating on a IoT ecosystem testbed to demonstrate secure routing of user data information between participating IoT devices showing efficiency of the proposed framework in terms of network stability, quality of services and performance metrics, such as network throughput, packet loss rate, latency rate, computational overhead using derived path loss and propagation model simulations.*

The rest of the paper is organized as follows. We critically review the background and related work in Section 3.2. In Section 3.3, we describe the proposed framework architecture. In Section 3.4, we defined the requirements for the implementation of the security framework. In section 3.5, we detail the implementation of the security framework. We discussed the security analysis of the proposed framework in section 3.6. We present the implementation, the results and discussions in section 3.7. Finaly, we summarized the paper in section 3.8.

3.2 Background and Related Work

3.2.1 Background

Researchers are increasingly interested in mobile payment and telecommunications environments that are becoming increasingly heterogeneous. This interest translates into the IoT Internet of

Things [43] and its heterogeneity yields to many fields of research covering payments with the Internet of Things (IoT) connected objects and blockchain [44].

However, the inherent payment success challenges such as [45]:

1. no connectivity;
2. authentication failure;
3. lack of secure user data routing;
4. lack of device security.

Those challenges demonstrate that there are contributions to be made to improve payments in highly heterogeneous environments such as *IoT* and *smart cities*.

Current explored research works lead to mention that the management of telecom services is influenced by a new paradigm of *software-defined networking* (SDN*)* allows the service provider to deploy scalable architecture providing flexibility in the management and control of services and terminals [46] ,47], [48].

In addition to being flexible, telecom architecture enables low-end terminals to have the power of calculations and physical resources and be less energy-intensive and more efficient. This is the ideal scenario for mobile payment to start adopting an architecture that supports another paradigm, such as *mobile edge computing* (MEC) [46],[49].

It is worth noting that the payments industry is full of very few essential players offering payment services at competitive costs while ensuring security and quality of services intimately associated with user requirements [49],[50].

Therefore, the network environment will integrate various IoT sub-systems, such as payment systems and payment applications which data routing is more prone to both distributed and non-distributed denial of service (DDoS/DoS). Such pretreated attacks could absolutely be harmful to the SDN network resources, forwarders, and controllers, where IoT user devices and transactions information details can be spoofed and forwarded to unauthorized IoT nodes. Moreover, the harmfulness of DoS attacks extends to smart IoT environmental applications and leads to a severely compromised network quality of service and performance [55].

As a result, we are focusing on implementing a new environment of mobile payment to support a variety of users' terminals of *IoT* when performing offline and online secured transactions based on the benefits offered by *software-defined networking* and *mobile edge computing*.

Securing mobile payment in *IoT* and *smart cities* may require understanding *IoT* environments in which mobile payment will be executed. In this regard, Ali *et al.* [45] explored solutions to meet IoT security requirements from the infrastructure, seeding communication between multiple sensors and monitoring applications. Threats in *IoT* and *smart cities* are at a new scale because of billions of intelligible connected things cooperating virtually in several unpredictable ways. Securing payment may involve an adapted network and protocol security for those billion sensors to securely communicate. Therefore, several resources must be made available to match the scale of architecture that connects billions of devices. The framework architecture's core features are presented as follows:

1. Data and privacy preservation anytime and anywhere in a cross-environment.
2. Identity management through myriad authentication schemes.
3. Trust and governance for fostering IoT into day-to-day user's life.
4. Fault tolerance to provide over-the-air updates and maintenance for billions of devices artificial intelligence (AI) readiness.

IoT devices are growing rapidly and are subjected to various security and privacy concerns when exchanging data with manufacturers, operators, and/or other connected devices. Sivaraman *et al.* [42] conducted various tests of safety and performances on consumers' *IoT* devices, then evaluated the results from manufacturers of *IoT* tested devices and proposed solutions for the identified security risks for consumers' *IoT* devices. *IoT* and *smart cities* environments favor the fast increase in devices and generate data. Therefore, researchers explore the relation between *IoT* and emerging technology, such as big data analysis, cloud and fog computing. Similarly, M.N. Bhuiyan *et al.* [43] presented a layered IoT architecture to embrace the IoT's security elements standards.

3.2.2 Identifying Threats to Existing Mobile Payment Environments.

As mentioned previously, several authors have proposed various solutions to secure mobile payments and user data. However, a quick analysis leads us to believe that some avenues of research are still not explored for securing payments in heterogeneous environments such as *smart cities* and *IoT*.

The related work shows mobile payments executed on different terminals or platforms, depending on the type of terminal chosen by the users. Thus, it is revealed that the cost associated with making mobile payments on the user side varies due to aesthetic preferences, technology, and convenience. In addition, the accessibility and availability of services should not be overlooked. The latter will be able to play a decisive role in the broad adoption of mobile payments. For this reason, it would be interesting to develop a payment system that would consider these different peculiarities [51].

However, the problem of securing mobile payments prevails. In addition, considering the context of *smart cities* and *IoT* with connected terminals, the process of securing mobile payment transactions and mobile payment users' data remains a relatively complex problem to consider and solve. Several researchers will focus on the issues that would cause obstacles in securing mobile payments in smart cities and IoT environments, as described below.

Attack and Threat Model

An adversary attacker initiates a request for services spoofing other users by opening a session with other legitimate user devices or terminals. The attacker can inject, replay, and eavesdrop on forwarded packets passing through controllers and forwarders. Therefore, attackers can succeed in impersonating a legitimate user. In this paper, we consider the entire possibility that the IoT device owned by a legitimate user can be subjected to physical attacks by an adversary attacker [52].

The attacker intends to succeed in bypassing the current authentication protocol by authenticating itself with the controller node server and masquerading as the legitimate user. More precisely, the attacker gains access to controlled forwarders and controllers without being detected. Therefore, in this paper, we focus on developing a framework that will authenticate the identity of the IoT device using the IoT device signal radio measurements to lock the authentication to a specific location point. Once the authentication is locked in, the framework deploys a secure routing channel to forward data in a secure way. Therefore, the proposed framework mitigates spoofing, cloning, physical attacks, etc.

Threats To the Mobile Payment Infrastructure Environment

First, we wonder about the problem of non-connectivity linked directly to the frequent partial or total network cut-offs or outages. In this regard, researchers should reconnect with the infrastructure during a partial or total outage. Although, researchers have no control over the choice of payment providers, mobile or configuration settings of users (i.e., transmission speed links, nodes interconnections or the maximum capacity of the latter), they are trying to specify the normal conditions for the execution of payment in connected and non-connected mode.

In a second step, researchers looked at the challenge of localization in IoT environment for mobile payment applications running on users' mobile devices. Indeed, the presence of several communication devices in the same environment brings up the problem of communication network congestion. Therefore, the related congestion problems of devices may provide different methods to solve these problems. However, real-time localization in a highly congested environment still makes the resolution of this problem more complex. Therefore, researchers should make a combination of approaches prioritizing the performance of devices and data security.

In addition to the complexity, mobile payment users could face the problem of a lack of identity tokens. The payment token can be in the form of hardware (i.e. traditional bank plastic card), or in the form of non-materialized or digital (i.e., cryptocurrency or mobile wallet). It indicates that a user could forget or even misplace a bank card, or the latter could be non-functional at a given time, which will prevent the user completing transactions at automated teller machines (ATM) or POS outlets.

However, these technologies depend on the functional connectivity of the provider's data network. Therefore, if there is a break in connection, the user will not be able to use the usual payment tokens. Therefore, it would be very wise to find a better solution that would allow the service of payment to succeed even in cases of loss of payment token or non-connectivity. To this end, it becomes essential that the mobile payment application executing on a terminal can retrieve, send, and receive location information regardless of connectivity infrastructure for performing outgoing and incoming financial transactions. Therefore, it is important that the operation and performance of mobile payments guarantee user data security in all use [53].

Another point of interest concerns the lack of privacy of users due to the class of service that users interact with by submitting data sensitive personalities through application interaction interfaces. Indeed, the lack of standards in the field of mobile payments and the monopoly of certain solutions belonging to manufacturers of mobile terminals or operating systems, generate inconsistencies in the practices used when processing transactions that would ensure keeping security standards. This can be explained by the fact that manufacturers of payment terminals can dictate which technology their terminals support or not and decline the compatibility of payment services with other technologies. Therefore, to better guarantee the security of user data, it would be reasonable to produce a validation of mobile payment security settings. Then, it is a question implementing the payment security mobile and evaluating any form of persistence of location data.

Threats To Mobile Payment Devices and Applications

A second concern is the expectations of securing users' mobile devices from software threats. It requires analyzing different types of attacks perpetrated against mobile payment users to strip them of their assets. Indeed, the research on the security of mobile payment systems identified risks such as the masquerading of users, malware, man attacks in the middle, and interception of traffic. In addition, the lack of two-factor authentication increases the risks of sending money to the wrong account or passing a password [55].

Financial institutions have responded to these threats by providing consumers and authorized service managers who approve and verify all mobile transactions [56]. This process is aimed at user customers and third parties to have confidence in the system and ensure the security of all partners. In addition, the sensitive nature of transactions between banks and mobile payment service providers requires that any conflicting issues be dealt with effectively to achieve the objectives [57].

However, it is crucial to mention that offline scenarios of mobile transactions are more difficult to get protected. In these cases, customer data is kept in the *POS* sales terminal for much longer, which is more exposed to attackers. Related location services in *IoT* and smart cities environments are good challenges for the sustainability of mobile payments. This extends further with the congestion of *IoT* devices that could potentially affect payments from the *IoT* Broadband networks. Hence, it implies the relevance of finding optimal mechanisms for implementing the collaboration of resources that can be used for protection by location awareness. While looking in a little more detail at the origin of mobile payment transactions, the lack of location of transactions would be at the expense of the security measures of initiated payments by users.

Security Requirement

This paper attempts to achieve the following objectives:

1. The authentication between an IoT device and an IoT network controller should be opened.
2. The location of the authenticated IoT devices should be known and unique to each IoT device.
3. An IoT device's memory should not leak any stored secret keys or tokens.
4. The proposed framework should not be working on compromised physically attacked IoT devices.

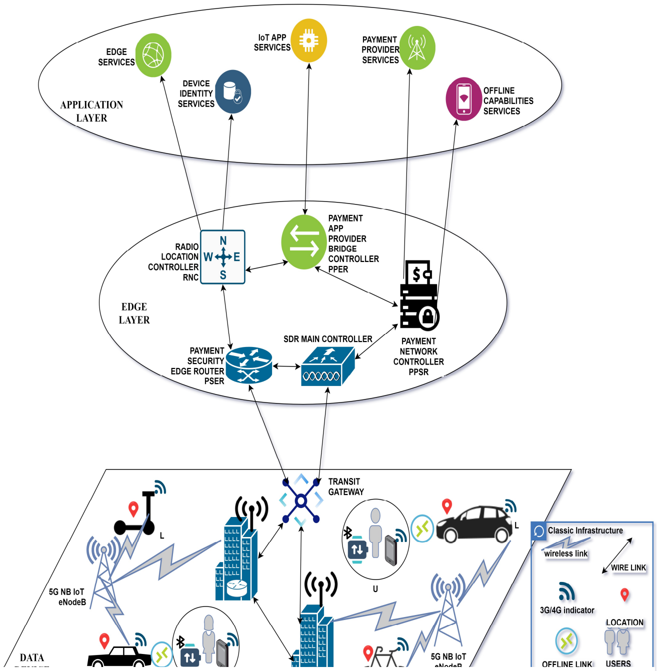


Figure 0.1 Proposed architecture of mobile payment.

3.3 Proposed Cross IoT Environment Framework Architecture

Our area of research targets security frameworks in IoT architecture and environments. Meanwhile, the proposed architecture illustrates *IoT* environments require it to be designed according to the security requirements for mobiles and payment transactions stated in Section 3.2.2.4. Therefore, securing applications and services in IoT environments requires a novel architecture. Thus, this paper proposed a novel architecture design, as shown in figure 3.1. More specifically, it offers a new model for IoT devices, applications, and services to enable online and offline secured transactions at any time in a dense environment with high connectivity and mobility. The proposed architecture is composed of three levels: application, edge, device. These three-layer levels communicate with service controllers at each level.

3.3.1 Architecture Description

3.3.2 Architecture's Main Communication Entities

The different entities involved in the proposed mobile payments architecture model are:

1. **Users** – *u*: User of mobile payment in InterCloud's environment;
2. **Users Group** - *U*: User and user's group of *IoT* devices;
3. **Device Location** – *L*: *IoT*'s device location at a given time;
4. **Payment Provider Services Router** – *PPSR*: Payment Service Router in the data layer;
5. **Payment Provider Edge Router** – *PPER*: Payment Edge Router in the edge layer;
6. **Radio Network Controller** – *RNC*: *SDN* main controller in the edge layer;
7. **Payment Security Edge Router** – *PSER*: *IoT* Security main Router in edge layer.

3.4 Security Trust Model

The need for the security trust model is crucial for identifying all interacting and communicating parties, confirming the attacker model, and setting the security goals.

3.4.1 Communicating Parties

1. **Users Group** - *U*: User and the user's group of *IoT*'s devices. Devices are IoT-based and are computationally limited devices such as a smartphone and a smart wearable band. The devices are usually called middle devices.
2. **Edge Network Controller** – *RNC*: *SDN* main controller in the edge layer. This is a more powerful device compared to the User group devices.
3. **Transit Network Controller** – *TNC*: *SDN* main controller in data and device's layer. This is a data device's forwarder and can be an embedded IoT device with more computation power than the user group device but lower than the *RNCs.*
4. **Application Network Controller** – *ANC*: *SDN* main controller in the applications and services layer. This is an application data devices forwarder and an IoT server with more computation power than the *RNCs.*

3.4.2 Attacker and Adversary Model

The attacker's model was discussed in section 3.2.2 and section 3.3.2. More specifically, the attacker has the following abilities:

1. Observation of the network activities
2. Alteration of data information
3. Causing delays on routing paths
4. Deletion of transmitted information between all communicating parties

The adversaries attacking the proposed framework system can be both internal and external. Attackers might have gained total control over the network among IoT devices, network controllers and forwarders. In the context of mobile payment, attackers are looking forward to launching fraudulent transactions on behalf of users. An IoT device may be lost.

3.4.3 Security Goals.

The security goals facilitate the understanding of what the proposed framework will ensure under the attacker model.

1. **Authentication** – Both the controllers and the IoT user devices can confirm the identity and legitimacy of the users.
2. **Confidentiality** – The only legitimate participating parties have the knowledge of the session key being used.
3. **Data forward secrecy** – In case of a compromised session or non-compromised previous known session, a key can be used. That will ensure data forwarding resiliency.

3.5 Security Framework

In this section, the security framework in Figure 3.2 of the architecture presents our approach of *IoT* context with security – authentication and secure data routing, assuming the lack of trust among heterogeneous IoT devices in the network while ensuring the existence of always continuation of service. For these reasons, in this paper, the core of the proposed architecture will be built on *mobile edge computing* (MEC) [46] and *software-defined network* (SDN) [49].

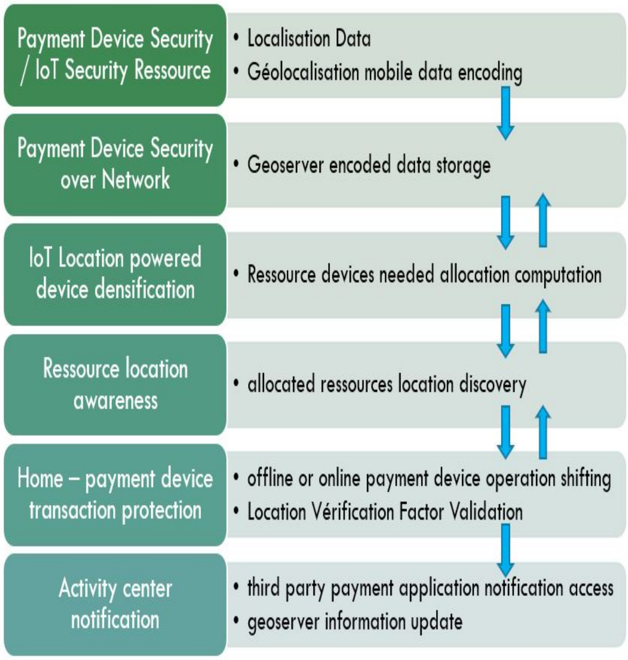

Payment Device Security / IoT Security Ressource
• Localisation Data
• Géolocalisation mobile data encoding
Payment Device Security over Network
• Geoserver encoded data storage
IoT Location powered device densification
• Ressource devices needed allocation computation
Ressource location awareness
• allocated ressources location discovery
Home – payment device transaction protection
• offline or online payment device operation shifting
• Location Vérification Factor Validation
Activity center notification
• third party payment application notification access
• geoserver information update

Figure 0.2 Proposed architecture's security framework for mobile payment protection.

This proposed security framework works as follows:

1. Authentication by location verification [53],[55]

- The geolocation data from the *IoT*'s terminal *GPS* and their encoding are converted in the form of geohash codes [58] for temporary recovery.
- The storage of geohash codes information is updated into the server for routing data to *IoT*'s terminals while roaming;
- The placement allocation of the IoT resources needed for enabling secure data routing.

2. Secure and trusted data routing [56],[59]

- Switching of trusted *IoT* terminals to an online mode or offline mode after checking the location network available gateways;
- Secure data routing transfer based on the cryptographic protocol RSA enforced by transit gateways and network controllers;
- Notifications of security status are sent to the requested IoT application or services listener, and updates are made on the forwarder server.

Notations: Table 3.1 contains annotations used in this paper.

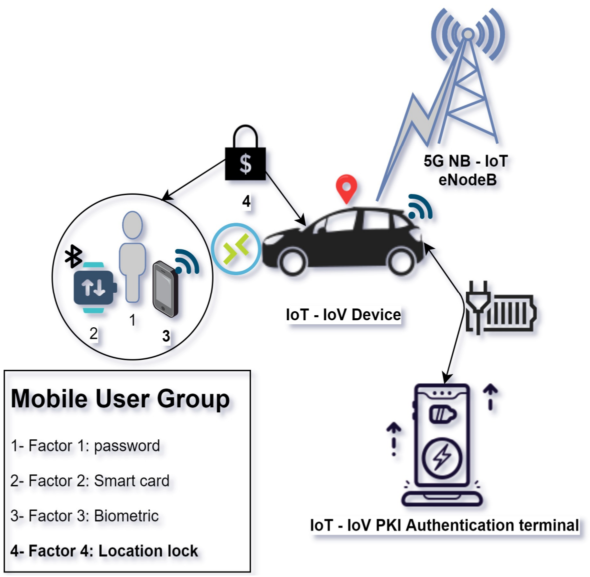


Figure 0.3 Proposed architecture's IoT location verification lock

Table 0.1 Algorithm Anotations

On	Source devices
Dn	Destination devices
Gn	Gateway devices
Pn	Path to be selected
R_{DL}	Downlink link Budget
P_{ALERT}	Probability of False Alarm
P_L	Propagation path loss
P_r	Received power

The choice for placement location of IoT resources participating in the security framework is justified using placement location algorithm responsible for making IoT resources available for *IoT* devices to perform secure data routing. The location verification lock is shown in Figure 3.3. This verification lock is made possible through an improved security network path selection procedure in Algorithm 1 [49].

3.5.1 Authentication Using Location Verification

Users of most applications, especially payment applications, will heavily rely on IoT devices. IoT device transceiver chipsets are equipped with RSSI measurement circuity for channel sensing.

The power received by the IoT device antenna can be measured using the free space path loss model given by:

$$P_r = P_t G_t G_r (\frac{l}{4\pi d})^2 \qquad (3.1)$$

ALGORITHM 1

IOT RESOURCES POSITIONING WITH SECURE PATH SELECTION

```
Input: S(V).
V = {On ∪ Dn ∪ Tn}, On= {O1, O2, O3...... om}, dn = {d1, d2, d3...... dm}, tn = {T1, T2, T3...... Tm}, Pn = {P1, P2, P3...... Pm}.

Output: Optimal selection of IoT resource with least expected time path and improved security

1: Procedure Initialization
2:   Create path .$p^{od}$
3: .  Set the time $\mathrm{T}(od) = 0$ spent for traversing $p^{od}$
4:   Calculate w(r) mean path and e() extension path.$p^{od}$
5:   Set  as maximum iteration with ${P^{od}}_{max} = \{p^{od}\}SEL_{max}={P^{od}}_{max}$.
6: Procedure Selection
7:   if  then $SEL_{max} = phi$stop;
8:   else continue step 5
9:   Select $p^{oi}$ from  and $SEL_{max}$for (i = 1; i ≤ max; i ++) do
10:      update ${SEL^{i}}_{max}$
11:    if i = s then stop
12:    else continue step 5
13: Procedure secured path selection
14:    Compute the new extended path  $e(p^{oj})$for j ∉ $p^{oi}$& $Pol^{od}$=$Pn^{od}$
15:    ;$Pn^{od} = \{Co, TC, SI_{i,j}, SR, MB, MA\}$
16:   if $PH_{i,j}$∈ $PH_i$then
17:        $IC_{i,j} = 1$;
18:        apply ;$SR_{i,j,l}$
18:    else $SI_{i,j}$+=;$IC_{i,j}$
19:   if  $S_l$∈ $PH_j$then
20:        SR + = ;$RU_{i,j,l} * IC_{i,j}$
21:   end if
22:   if  then$M_l$ ∈ $PH_{i,j}$
23:       MB+= ;$IC_{i,j}$
24: MA+= ;$MB_{max}$
25:    end if
26:  end if
27:  return the optimal IoT Resource secure selection $SI_{i,j} \geq Cov$
28: end
```

Equation (3.1) can be expressed as the logarithm distance path loss model in (3.2):

$$P_r(d)[dBm] = \bar{P}_r(d) + X_S + X_R, \qquad (3.2)$$

where is the average received power at distance $\bar{P}_r(d)d$.

We assume the following:

$$X_S + X_R{=}X_\sigma, \qquad (3.3)$$

where is a normally distributed variable taken random, describes the shadow effect taken random, and captures the effects of the Rayleigh fading [53].$X_\sigma X_S X_R$

From (2), can be expressed as the following:$\bar{P}_r(d)$

$$\bar{P}_r(\mathrm{d}){=},\bar{P}_r(d_0) - 10n_p\log\left(\frac{d}{d_0}\right) \qquad (3.4)$$

where is the average received power at a reference distance , and $\bar{P}_r(d_0)d_0n_p$∈ [2,6] is the path loss exponent depending on the environment of the transmission media.

Therefore, with = 1 m, (3.2) can be rewritten as: d_0

$$P_r(d)[dBm] = \bar{P}_r(1) \text{ - } + 10n_p\log(d)X_\sigma \qquad (3.5)$$

We assume that an IoT device is originally located at d distance from the signal transmitter. This assumption let us represent a probability density function (PDF) of the received power of the IoT device compared to a fixed IoT node using its original position expressed as $F_d(P_r[dBm])$.

Therefore, the false alert denoted for an IoT device with a fixe PDF is calculated as: P_{ALERT}

$$\int_{-\infty}^{\gamma_{treshold}} F_d\ (P_r[dBm])dP_r = P_{ALERT} \qquad (3.6)$$

In the advert of an IoT node or device attacked, or moved from its original location, then the measure received powered is given as follow:

$$Loc_{STATE}= \begin{cases} \text{"no change"}, if\ and\ only\ if\ P_r[dBm] \geq \gamma_{treshold} \\ \text{"compromised"}\ ,\ \ attack\ including\ location\ change \end{cases} \qquad (3.7)$$

Figure 3.4 depicts the framework workflow of the authentication verification.

- **Authentication model framework description**

As shown in Figure 3.4 and table 3.2, the proposed authentication scheme is detailed as follows:

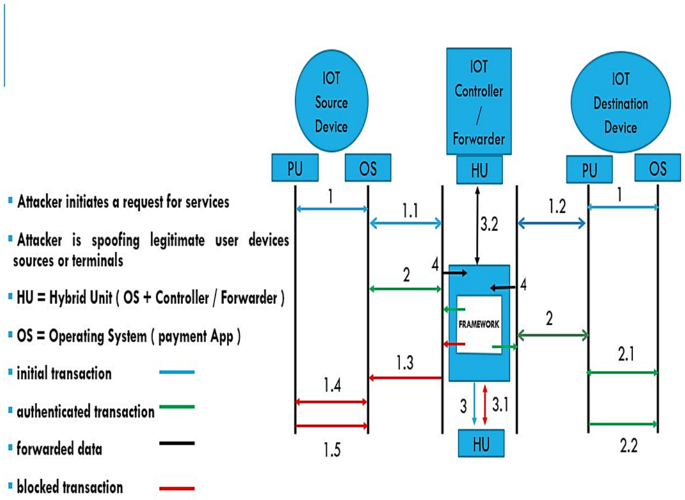

IOT Source Device
PU
OS
IOT Controller / Forwarder
HU
IOT Destination Device
PU
OS
Attacker initiates a request for services
Attacker is spoofing legitimate user devices sources or terminals
HU = Hybrid Unit (OS + Controller / Forwarder)
OS = Operating System (payment App)
initial transaction
authenticated transaction
forwarded data
blocked transaction
1
1.1
3.2
1.2
1
2
4
4
FRAMEWORK
2
1.3
2.1
1.4
3
3.1
1.5
HU
2.2

Figure 0.4 Proposed framework workflow activities for authentication

i th User – IoT source device - ,U_{S_i}

k th IoT controller device – ,P_k

j th User – IoT destination device - ,U_{R_j}

- **STEP 1**: sends to initiate an authentication request to the corresponding IoT service gateway that checks identification;$U_{S_i}NID_{S_i}$ $P_kU_{S_i}$

- **STEP 1.1**: **If** device identification is confirmed **then** the IoT service gateway P_kComputes NID_{G_i}**;**

- **STEP 1.2**: sends M_{G_i} to the corresponding IoT device ;U_{S_i}

- **STEP 1.3**: **else** requested authentication is aborted;

- **STEP 2**: sends to the corresponding IoT service gateway using U_{R_j} NID_{S_i} M_{R_i};
- **STEP 3**: The service gateway samples RSSI values for , and itself and saves it as a socket as a fingerprint of the authentication request;$P_k U_{S_i} U_{R_j}$ $P_k SOC_{rssi} = \{U_{S_{rssi}}, U_{R_{rssi}}, P_{k_{rssi}}\}$
- **STEP 3.1**: **If** **then** $SOC_{rssi} \leq Loc_{STATE}$requested authentication is aborted;
- **STEP 3.2: else** requested authentication is completed;
- **STEP 4**: The service gateway decrypts $P_k M_{R_i}$ to confirm device U_{R_j} identification;
- **STEP 5:** Session key is computed.$K_{S,R,G}$

Table 0.2 Authentication Scheme Annotation

S_i, P_i, R_i	Ban logic principals Source (S), temporary destination (P), Receiver (R)
N_{device}	IoT device nounce number
ID_{device}	IoT device identification number
NID_{G_i}	$NID_{G_i} = \{,\} N_{Gi} ID_{Gi}$
NID_{S_i}	$NID_{S_i} = \{,\} N_{Si} ID_{Si}$
M_{device}	message sent by IoT device
M_{G_i}	$M_{R_i} = \{ID_{S_i}, N_{S_i}, N_{P_k}\}$
M_{R_i}	$M_{R_i} = \{ID_{S_i}, N_{S_i}, N_{R_j}\}$
K_{device}	Session key for the IoT device
H()N_{device}	Hash encryption function
$K_{S_i R_j}$	$K_{S_i R_j}$= H() N_{S_i}⊕ H()N_{R_J}

3.5.2 Secure Data Forwarding Using a Secure Route with Trusted Data Routing

We are assuming that in the proposed framework, the IoT devices are mobile. The IoT devices change their positions at each instance of time and space. Therefore, the final position of the IoT devices can be computed using the Euclidean function, denoted by:

d(x,y)=,$\sqrt{(x_d - x_i)^2 + (y_d - y_i)^2}$ (3.8)

where () be the initial position and () is the request destination position of the IoT device.$x_i, y_i x_d, y_d$

In (3.8), we can express the packet loss rate model (PLR), which is designated by:

P_{LR}=$R_x(t_0)/T_x(t_0)$ (3.9)

From (3.9), we can express the link cost (LC) as:

$$L_{COST}=1-P_{LR} \qquad (3.10)$$

The source IoT device will reach the appropriate next hop for data routing whenever the has the least calculated value.L_{COST}

(3.10) allows targeting a fail-free, robust delivery ratio.

ALGORITHM 2

IOT NETWORK CONFIGURATION

```
Input variables: $Nn = \{N1, N2, N3...... Nm\}$, $Gn = \{G1, G2, G3...... Gm\}$, $Tn = \{T1, T2, T3...... Tm\}$, $Pn = \{P1, P2, P3...... Pm\}$, $Rn = \{R1, R2, R3.... Rm\}$,
Output: Netconf File
1: procedure Network Configuration
2:   Read positions IoT Participating devices.
3:   and   Set the time $\mathrm{T}(config) = \varphi$ spent for reading $p^G \leftarrow Gnp^N \leftarrow Nn$
4:   Read routing path $R^N$ and $R^G$
5 :  Calculate IoT source device distance – vector d(x,y)
6:   Based on $\Delta T(config), d(x,y) = \sqrt{(x_d - x_i)^2 + (y_d - y_i)^2}$
7:   for $Nn \in [1, Nm]$ do
8:       testing $L_{COST}$ $AND$ $1 - P_{LR}$
9:           if   where $L_{COST} = 1 - P_{LR}P_{LR} = R_x(t_0)/T_x(t_0)$
10:      do Nm select least  with datalink available for transmission$LC_{LINK}$
11:    update routing path $R^N$ and $R^G$
12:    end if
13: end for
14:   for $Gn$ $[1,Gm]$ $\in$do
15        if   then $p^G \equiv p^N$
16:            Direct Data Forwarding
17:       end if
18:       go to step 8
18:    until   $p^G! = p^N$
19:   end for
20: end procedure
```

The proposed scheme goal is to establish a reliable routing path between the IoT network controller and data forwarder. Therefore, Algorithm 2 describes the formation of the network setup with a reliable data routing path among IoT source devices and IoT destination devices. IoT source devices and destination devices can be either IoT network controllers or data forwarders, such as IoT gateways and IoT routers. However, IoT source devices are IoT wearable devices and IoT mobile user equipment, such as smartphones, smartbands, etc.

Practically, when the next hop is selected, the IoT source device unicasts an RREQ message and in response, the next hop replies with an ACK message. The same process is repeated until a reliable and secure path is established between the IoT source devices towards the IoT gateway.

The proposed scheme will secure the selected path by using a technique of cryptography. The proposed technique presents a fully secure and trusted data routing scheme following an asymmetric-based RSA cryptographic technique implemented as public-private keys adopted in [56].

In this cryptographic technique, the IoT source device needs to encrypt and decrypt incoming messages from the neighboring IoT destination device. Therefore, each IoT source device possesses a pair of a public key (PuK) and a private key (PvK). The PuK is a distributed key from a secure IoT service network router and the is used to receive encrypted messages from IoT neighboring devices. The PvK, known as the secret key, is used to decrypt the incoming message toward the IoT source device. PuK_{Device}

The data authentication is performed by the use of .PuK_{Device}

Thus, they must be kept secure from malicious and unauthorized IoT devices.PuK_{Device}

The network topology used in this proposed technique is fully connected. The advantage of the fully connected network topology is that IoT participating devices use the public-private keys to send encrypted messages and decrypted received messages. For example, E(m) represents the encrypted message m from IoT source device X to IoT destination device Y, using Y known public key . PuK_Y

Upon reception of E(m), IoT device Y, Y decrypts E(m) using Y private key .PvK_y

Since the message was broadcast toward all other available IoT devices, including Y, only Y can decrypt the message, whereas other IoT devices cannot. At the same time, an acknowledgment ACK is sent to the corresponding IoT source device to confirm the reception of the message message m.

In addition, the proposed security technique preserves data security by requesting robust authentication between all IoT participating devices. Therefore, algorithm 3 finalizes the secure and trusted data routing between authenticated IoT participating devices.

ALGORITHM 3

TRUSTED AND SECURE DATA ROUTING

```
Input variables: $PuK_i$,$PvK_i$, i ∈ {$X$, $Y$}
procedure trusted and secure data forwarding
  Read positions IoT destination forwarder devices.
  GOTO Algorithm II for building routing paths
    extracting routing paths for IoT service RSA routers
  IoT RSA router () sharing for IoT source devices $K_i PuK_i, PvK_i X_{device}, Y_{device}$
  IoT $X_{device}$ generates $m$
  E(m) = $H$(m ⊕ )$K_X$
  IoT $Y_{device}$ receives E(m) sent by $X_{device}$
  IoT $Y_{device}$ decrypts E(m) using $PvK_Y$
  m = $H$(E(m) ⊕ )$K_Y$
  IoT $Y_{device}$ sends ACK to IoT source $X_{device}$
  IoT service router update IoT network forwarders
  IoT network forwarder update IoT network controllers
end procedure
```

3.6 Security Analysis

In section 3.5, we listed the security goals that will be achieved by our proposed framework. This following section illustrates how the goals are achieved.

3.6.1 Formal Security Analysis

We mention that the secrecy properties are established by successful authentication using () for $N_{S_i}, N_{R_j} i, j \in \{1, netlimit_{\infty}^{CAP}\}$.

We begin this section of formal security analysis by illustrating the framework messages for authentication in Figure 3.4 as follows:

1. $S_1 \rightarrow P_I$: ,.$S_1 N_{S_1}$
2. $R_1 \rightarrow P_I$: ,.$R_1 N_{R_1}$
3. $P_1 \rightarrow R_1$: .$\{R_1, N_{R_1} \equiv N_{S_1}\}$
4. $P_1 \rightarrow S_1$: .$\{S_1, N_{S_1} \equiv N_{R_1}\}$
5. $R_1 \rightarrow S_1$: .$[\{1) \rightarrow 4)\} \cap \{\{2) \rightarrow 3)\}\}]$

The statements above are derived from Mao and Boyd's logic [60] to present a formal security analysis of our proposed security framework. The logic of Mao and Boyd helps to make the following assumptions:

1. $S_1 \sqsupseteq S_1 \xleftrightarrow{OTP^{1,1}} P_1$ and and , where identifies equipment identification (EmID) for each IoT device using a time-based one-time password algorithm (TOTP) [54].$P_1 \xleftrightarrow{OTP^{1,1}} R_1 R_1 \xleftrightarrow{OTP^{1,1}} S_1 OTP^{1,1}$
2. $S_1 \sqsupseteq R_1 \dashv N_{R_1}$ and $R_1 \sqsupseteq \sqsupseteq \dashv S_1 \{R_1\} N_{S_1}$: generates .$S_1 N_{S_1}$
3. $S_1 \sqsupseteq \dashv R_1 \neq \{\}$ and $N_{R_1} S_1 \sqsupseteq \dashv R_1 \neq \{\}$: generates new each time if needed.$N_{S_1} S_1 N_{S_1}$
4. $R_1 \sqsupseteq \dashv S_1 \neq \{\}$: generates new each time if needed.$N_{R_1} R_1 N_{R_1}$
5. $S_1 \sqsupseteq$ *nxt()*R_2 : is new that generates new overwriting previous .$R_2 R_1$ $N_{R_2} N_{R_1}$

6. $R_1 \sqsubseteq nxt()S_2$: is new that generates new overwriting previous .$S_2 S_1$ $N_{S_2} N_{S_1}$

Table 0.3 Security Proofs

$\dfrac{[S_1 \sqsubseteq S_1 \overset{OTP^{1,1}}{\longleftrightarrow} P_1\,] \cap [S_1 \overset{OTP^{1,1}}{\longleftarrow} N_{S_1} \equiv N_{S_2}]}{S_1 \sqsubseteq S_1 \overset{OTP^{1,1}}{\Longleftrightarrow} N_{S_1}}$ R_1 believes sent using as the encryption key$S_1 N_{S_1} OTP^{1,1}$
$\dfrac{[S_1 \sqsubseteq S_1 \overset{OTP^{1,1}}{\longleftrightarrow} P_1\,] \cap [R_1 \overset{OTP^{1,1}}{\longleftarrow} N_{R_1} \equiv N_{R_2}]}{R_1 \sqsubseteq R_1 \overset{OTP^{1,1}}{\Longleftrightarrow} N_{R_1}}$ $\boldsymbol{S_1}$ believes sent using as the encryption key$\boldsymbol{R_1 N_{R_1} OTP^{1,1}}$

7. S_1 : Step 2 of the framework is realized.$\overset{OTP^{1,1}}{\longleftarrow} N_{S_1} \equiv N_{S_2}$

8. R_1 : Step 3 of the framework is realized.$\overset{OTP^{1,1}}{\longleftarrow} N_{R_1} \equiv N_{R_2}$

9. S_1 $\sqsubseteq$ $\sqsubseteq R_1$ $\{\}S_1 \Dashv N_{S_1}$and $R_1 \sqsubseteq$ $\Dashv S_1 N_{S_1}$: generates new each time if needed.$R_1 N_{R_1}$

10. $R_1 \overset{OTP^{1,1}}{\Longleftrightarrow} N_{R_1}$: Step 2 of the framework is realized.

11. S_1 : Step 3 of the framework is realized.$\overset{OTP^{1,1}}{\Longleftrightarrow} N_{S_1}$

The logic table of Mao and Boyd [60], as illustrated in Table 3.3 proves the authentication properties of the proposed framework.

3.6.2 Protection Against Spoofing Attacks

An attacker or a ‘spoofer’ is a malicious participating IoT device or user that impersonates another device or user to bypass access controls while compromising network hosts, stealing data, and spreading malware. The situation is even worse since more IoT devices and terminals are easy to access by any user. Therefore, the attacker or adversary could clone the IoT device or terminal by extracting the stored secret keys from its storage memory drive. The proposed framework solves these issues using two combined features. First, the use of the RSSI-based location verification for each IoT participating device eliminates, validates, and ensures of the authentication legitimacy of the IoT participating device. This results in security against physical attacks perpetuated on IoT devices or terminals attempting to clone [61] the IoT device or making unauthorized use at different locations from the known location [62].

Moreover, the use of RSA, as the second feature of the proposed framework, aims to secure a trusted user data routing channel for IoT device communication. In this situation, data confidentiality and data forward secrecy are preserved in a preferred SDN-based IoT system.

3.6.3 Protection Against Dos Attacks

An attacker or adversary activities may gain access to vulnerable participating IoT devices and may try to break the communication between IoT sources devices and IoT destination devices. The adversary's main goal is to disturb the synchronization between the IoT device and IoT controller or data forwarder server by blocking packets carrying protocol parameters, as shown in Step 1.2 of Figure 3.4 step 1.2. Moreover, the adversary may obfuscate the exchange of data between the source IoT device and the destination IoT device by dropping packets carrying routing and security parameters, as shown Step 2 of Figure 3.4.

Therefore, in our proposed framework, the sources and destination devices use the for $OTP^{i,j}i\ , j$ $\in \{1, netlimit_{\infty}^{CAP}\ \}$ that identifies Equipment Identification (EmID) each time a request of authentication is sent by the source IoT device.

3.7 Implementation, Results and Discussions

In this section, the experiments are conducted for the proposed framework. The results are discussed in the following section with respect to different security goals, performance improvement indicators and network parameters.

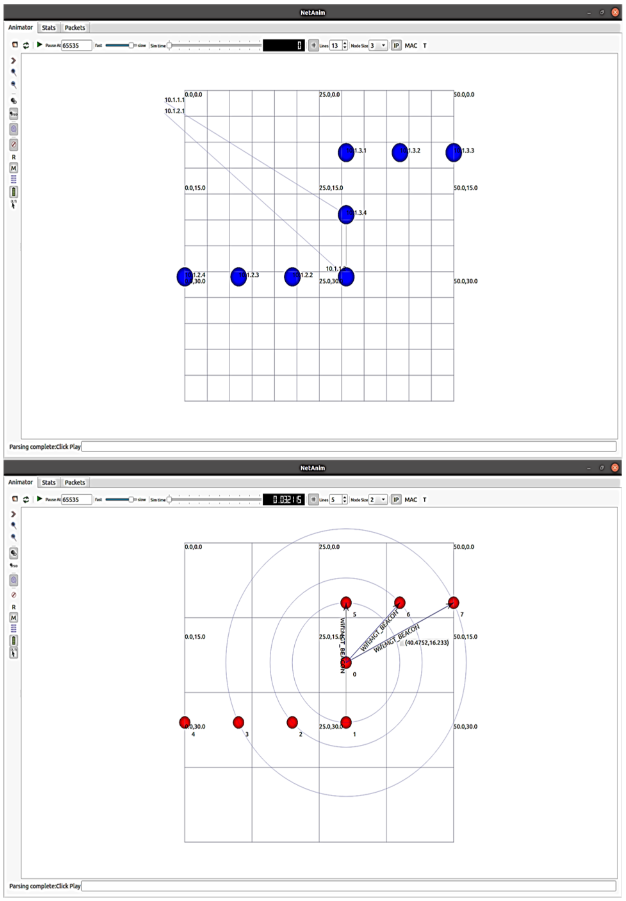
3.7.1 Experiment

To verify the theoretical analysis, we design related experiments to deploy the IoT system environment.

The environment is an SDN-based IoT environment. The results obtained reflect that most of the heavy computation is performed at the edge plane level to minimize the resource constraint of IoT participating devices located at the data plane level. In the IoT system environment considered for the proposed framework, controllers and forwarders known as destination IoT devices are allocated on a laptop and the middle device, known as the source IoT device, is configured on a virtual Linux machine.

In the experiments conducted, following SDN-based paradigm, POX [63] is used as an IoT network controller and Open vSwitch [64] as a SDN switch. iPerf [65] is used to emulate different network traffic matrices across individual fast Ethernet speed bandwidth of 100 Mb/s.

The proposed framework makes use of algorithm 1 to enable IoT resource placement with secure path selection.



Pause At 65535

Lines 13

Node Size 3

Parsing complete:Click Play

Pause At 65535

Lines 5

Node Size 2

Parsing complete:Click Play

Figure 0.5 NS3 framework Network implementation animation

Table 0.4 Network Set of Factors

Factors	Value
covered area	100 m – 500 m
deployment_size_mode	medium - random
number_of_node	50 - 500
number_of_malicious	2 - 50
max_messages (per node)	1 - 3
max_bulk_bytes	512 kb – 1024 kb
ddos_rate_CBR	100 mbps
max_simulation_time	1000 s
RSA_key_size	32 bits – 64 bits
MAC_layer	IEEE 803.X

The proposed scheme goal is to establish a reliable routing path between IoT devices, network controllers and data forwarders. Therefore, the proposed framework makes use of algorithms 2 and 3 to set the formation of the network setup with a secure data routing path among IoT source devices and IoT destination devices.

Therefore, the evaluated metrics are discussed in the next section.

Table 3.4 illustrates the default network set of factors used in conducting the experiments.

More details regarding the implementation configuration are illuminated in Table 3.5.

Table 0.5 Details about Experiment Device Configuration

IoT Device's Role	Environment	Details	
Source	Virtual Machine	CPU	1.5 GHz 64 bit
		Memory	1GB
		Storage	32GB
controller	Virtual machine	CPU	2.5 GHz 64 bit
		Memory	4GB
		Storage	64GB
destination	Virtual Machine	CPU	1.5 GHz 64 bit
		Memory	1GB
		Storage	32GB
forwarder	Virtual Machine	CPU	3.0 GHz 64 bit
		Memory	6GB
		Storage	128GB

First, the proposed framework is aligned to be verified with security architecture for attack detection and authentication [76], [39]. Second, the proposed framework is verified with wormhole-free routing Dos attack defense solutions [67].

In the simulation environment, IoT devices, gateways, controllers, and forwarders range from 50 to 500 devices clients deployed. The number of potential malicious IoT devices range from 5 to 50 with data traffic flow equivalent to a constant bit rate (CBR).

All the experiments are implemented using network simulation tool (NS3) [68], as shown in figure 3.5. The proposed framework is evaluated with the following metrics in terms of network throughput, latency rate, packets lost rate and computational overhead.

The implementation experiments were carried out in three separate simulated environments. We decided to evaluate the proposed framework, the RSSI value, which can be converted to the IoT device received power (), is set using the following analytical formula [69]:$P_r\ in\ dBm$

$$P_r = DEVICE_{Rssi} - \varphi, \qquad (3.11)$$

where $\varphi \in \{|-45|, 45\}$

Therefore, the simulation environments are described as

1. Indoor environment non–urban;

2. Indoor environment non-suburban;
3. Outside environment non-rural.

The indoor environment is defined by the distance ranging from 1 m to 10 m between IoT device transmitters. The linear model of the a and the logarithm of distance d is resulting from the value for fitted with a linear curve to estimate the value of and [70], readjusted with R-squared representative value of accuracy to the fitted model [71]. $P_r P_r n_p X_\sigma$

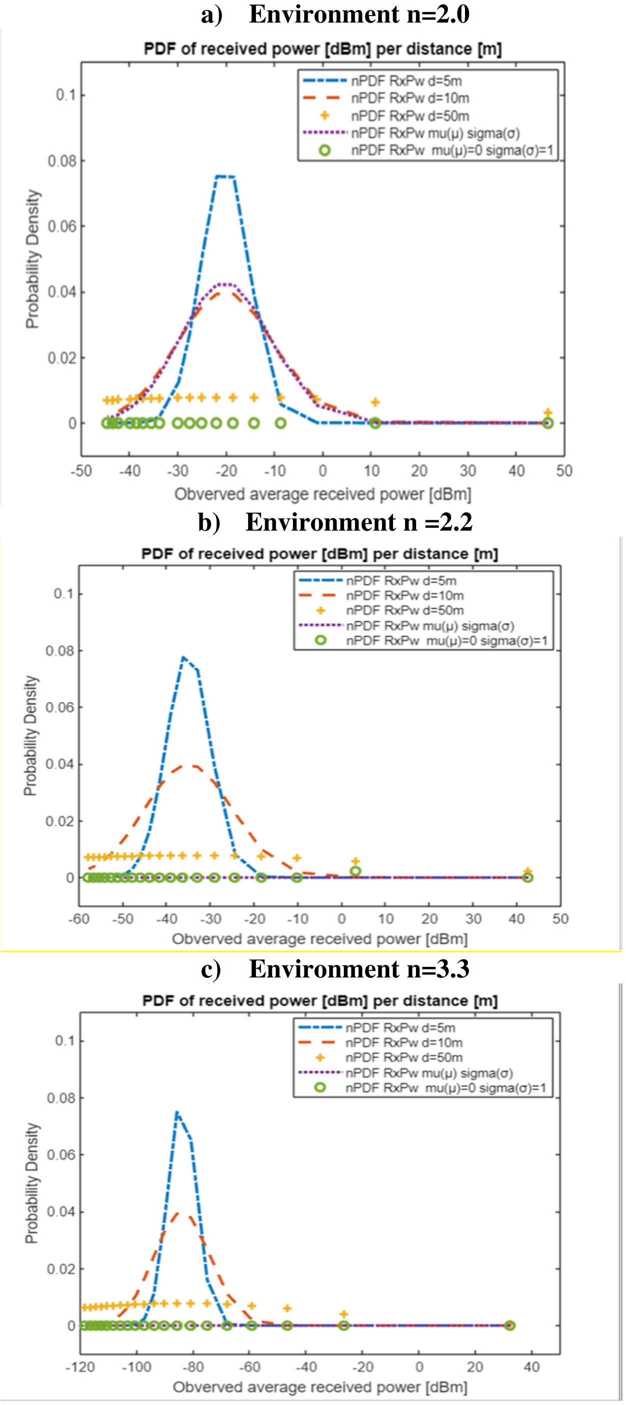
3.7.2 Results

In this section, we present the performance results using analytical models satisfying the properties of path loss, shadowing, and Rayleigh fading. The environment models are based on path loss models in the proposed architecture environment for the determination of the signal strength concerning various frequency ranges and distances for the considered IoT-based network. Thus, we considered three path loss models [72], namely Friss Free Space, Log-distance and Log-normal models. In the context of simulating an IoT environment, we derived the analytical models from the log-distance model. These models have been reviewed with different IoT receiver antenna heights in mainly urban and suburban environments. The respective derived propagation models for the three environments are described in Table 3.6.

Table 0.6 Environment Analytical Models

Environments ($d_0 = 1m$)	**Model**	**Path Loss Exponent (PLE)**	**P_r from Equation (1)**
Indoor environment non – urban	-20.112 log (d) -50.262	2.00	-46.67
Indoor environment non - suburban	-21.096 log (d) -70.012	2.2	-42.60
Outside environment non - rural	-30.063 log(d) -45.567	3.3	-32.45

- From equation (3.1), the *propagation path loss* in free space is denoted as *PL* and designated by:



a) Environment n=2.0

b) Environment n =2.2

c) Environment n=3.3

Figure 0.6 Average received power per distance for each scenario in Table 3.6

$P_L(\text{dB}) = -10 = +20\ log_{10}\left[\frac{\lambda^2}{(4\pi\lambda)^2}\right] log_{10}\left[\frac{4\pi d}{\lambda}\right]$(3.12)

- The propagation path loss at any distance $d > d_o$ is given by:

$[P_L(d)]_{dB} = [P_L(d_o)]_{dB}+10nlog_{10}(\frac{d}{d_o}) + \chi$ (3.13)

for

$d_f \leq d_o \leq d,$

where***x*** is a zero-mean Gaussian distributed random variable with standard deviation $\boldsymbol{\sigma}$ = χ = *2 dB*, $\boldsymbol{d_f}$ distance in the far field, is the path loss at an arbitrary distance$[\boldsymbol{P_L(d)}]_{\boldsymbol{dB}}$ *d*meters, ***n***is the *path-loss exponent*(PLE) with *PLENARY n* = **2**, and **=** 1 *m to 1000 m*.

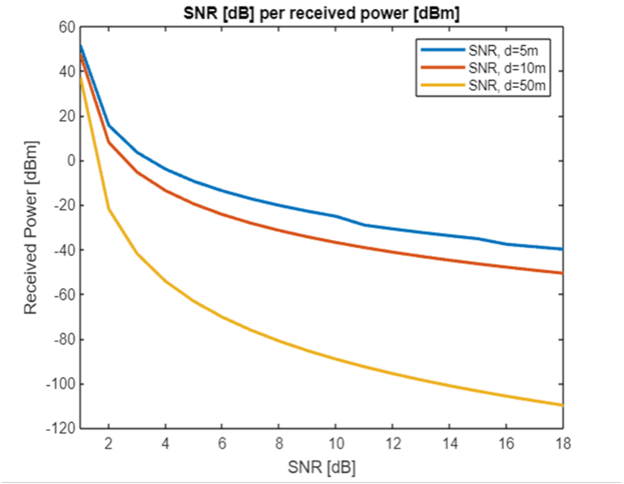


Figure 0.7 Probability of detection for each scenario in Table 3.6

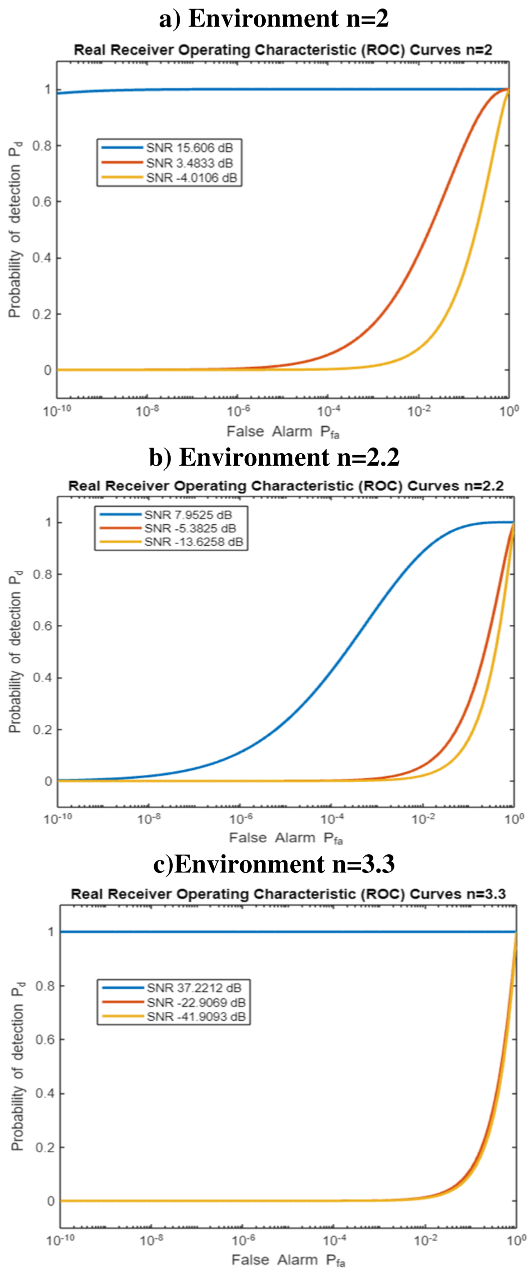
From (3.12) and equation (3.13), the generic path loss representation and power received between distance *d* varying from 1 *m* to 1000 *m* (*1Km*) without simulation yet.

After the simulation is complete using the set of parameters from our framework and data collected from NS3, Figure 3.6 illustrates the RSSI value converted into power received, which confirms how the signal power becomes non-detectable when the distance increases.

Figure 3.7 illustrates the probability of detection (*Pd*) of the emitted signal when the device did not cross the threshold of the false alarm *(Pfa)*. It is also worth noticing that the signal noise ratio SNR (dB) was evaluated for each environment to be able to draw the Real Receiver Operating Characteristics (RROC) for the device running our proposed framework set of parameters.

Thus, Figure 3.8 shows the signal noise ratio (SNR) calculated from the Received Power data collected. This graph illustrates the SNR that exists when devices are dense at a close distance from each other. The results in network performance degradation increase the risk of failure to deliver on network QoS, request services for participating devices, such as authentication and validation of packets, and find the best route for data.

Therefore, the next section presents the impact of the number of IoT network devices and malicious nodes on the network, followed by the discussion on the performance of our framework.


a) Environment n=2

b) Environment n=2.2

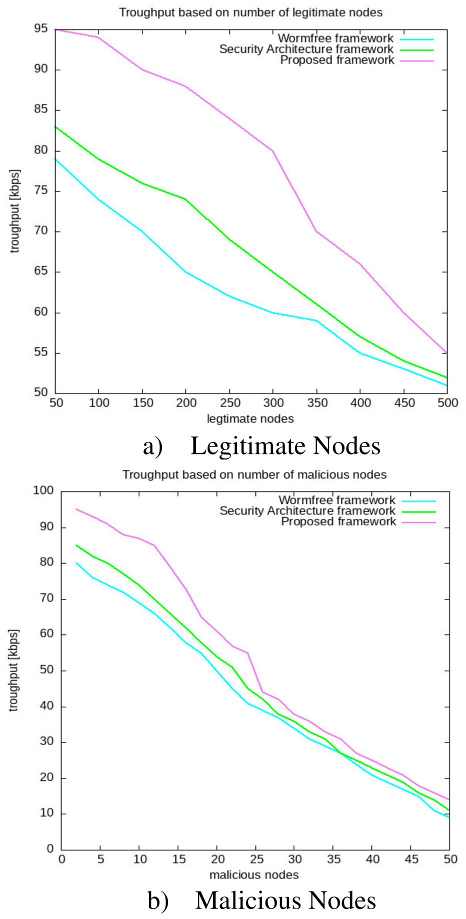
c)Environment n=3.3

Figure 0.8 SNR [dB] per received Power for d=5m, d=10m, d=50m

3.7.3 Discussions

In this section, the evaluated network metrics for the experimental configurations are discussed below.

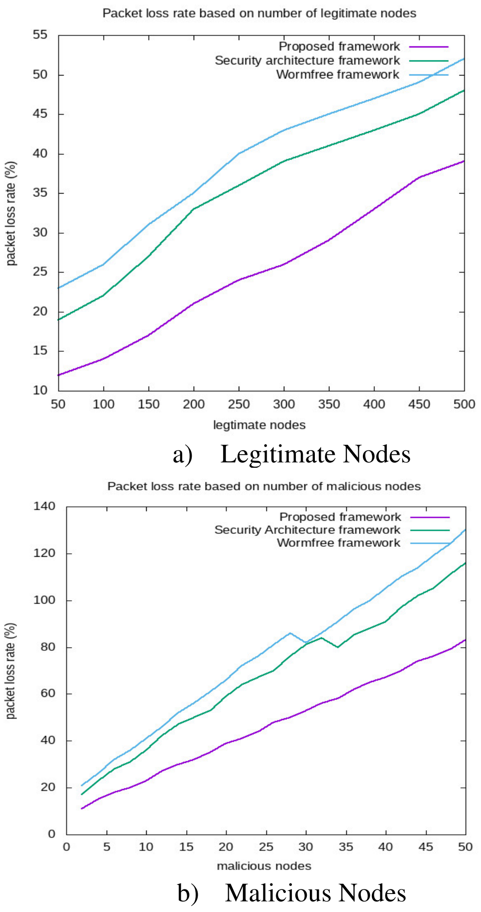
1. ***Positioning:*** In the proposed framework, the IoT participating device positioning is a placement Algorithm 1 used to find the positioning of IoT participating devices. Achieving the best positioning for IoT notes has an impact on network throughput, as shown in figure 3.9, and on packet loss rate, as depicted in figure 3.10. The proposed framework increased the network throughput by 13% and decreased the data loss rate by 35%. This can be explained by the IoT participating device selecting trustworthiness and stable routing paths.



a) Legitimate Nodes

b) Malicious Nodes

Figure 0.9 The impact of the number of nodes on network throughput



a) Legitimate Nodes

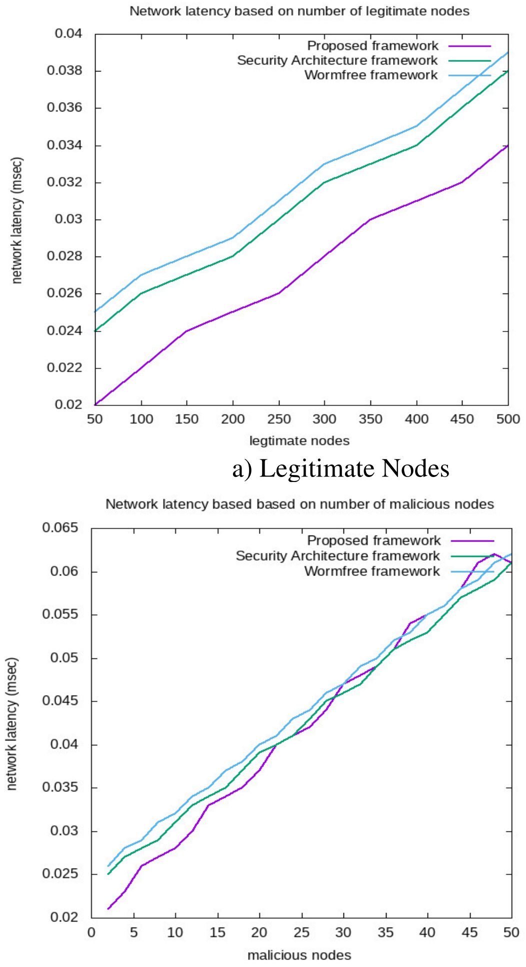
b) Malicious Nodes

Figure 0.10 The impact of the number of nodes on the packet loss rate.

2. ***Security mechanisms:*** In the proposed framework model is built using Algorithm 2. This latter aims to improve network stability and security. In case of any traffic overload, Algorithm 2 should reshape and protect the network in time, so that the IoT system network

could continue to run safely and stably. Figure 3.11 depicts that the proposed framework achieves the lowest latency and Figure 3.12 presents the computation overhead performance of our proposed framework. Our proposed framework decreases the network latency rate by 12.21%. This indicates that the routing of this higher IoT network topology is continuously adaptative due to the secure data forwarding features of our proposed framework.

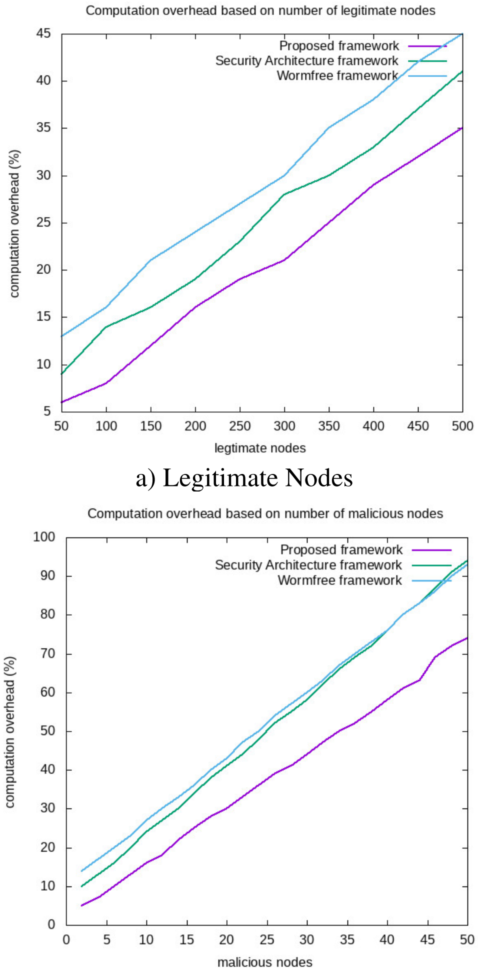
3. ***Resilience against security:*** In Table 7, we compare the properties and features for performance of the proposed framework with five existing applications centric security



a) Legitimate Nodes

b) Malicious Nodes

Figure 0.11 The impact of the number of malicious nodes on latency rate



a) Legitimate Nodes

b) Malicious Nodes

Figure 0.12 The impact of the number of malicious nodes on computational overheads.

schemes. Furthermore, the proposed framework in support of RSA public-private key encryption has an impact in reducing data re-transmission and re-routing in the simulated environment, as well as reducing network delay significantly.

Table 0.7 Features Comparison with other Proposed Framework

Evaluated security requirements	[7]	[35]	[41]	[42]	[43]	**OURs**
DDoS attack detection	And	N	N	N	And	And
Internal attack detection	And	And	N	N	N	And
Cloned App/device attack	N	Y	Y	Y	N	Y
Smart Card / Card not present Attack	N	N	Y	Y	N	N
Authentication security mechanism	N	N	Y	Y	Y	Y
Location / Positioning / Verification / detection	N	N	N	N	N	Y
Network metrics improvement	Y	Y	N	N	N	Y
Secure Data Routing	N	N	N	N	N	Y
Impersonation attack	N	N	Y	Y	N	Y

4. ***Computational Cost complexity:*** In Table 3.8, we present the complexity of the computation that is evaluated using the number of hash functions (), encryption/decryption (), modular

Table 0.8 Comparison of Computation Complexity

Framework	IoT Source	Server/Controller	IoT Destination
[39]	$4Nn_H+4N_{ECC}$	$3Nn_H+4N_{ECC}$	$4Nn_H+4N_{ECC}+N_{DEST}$
[67]	$4Nn_H+3N_{ECC}$	$4Nn_H+4N_{ECC}$	$4Nn_H+3N_{ECC}+N_{DEST}$
[72]	$2Nn_H+2N_{exp}+N_X$	$2Nn_H+2N_{exp}+N_X$	$2Nn_H+2N_{exp}+N_{DEST}$
[73]	$3Nn_H+4N_{ECC}$	$4Nn_H+4N_{ECC}$	$3Nn_H+4N_{ECC}+N_{DEST}$
[75]	$7Nn_H+6N_{ECC}$	$9Nn_H+7N_{ECC}+3N_{DEST}$	$7Nn_H+6N_{ECC}+N_{DEST}$
OURs	$1Nn_H+2N_{RSA}$	$1Nn_H+1N_{RSA}+2N_{TRX}$	$1Nn_H+1N_{RSA}+1N_{RX}$

multiplication (), modular exponentiation (), point addition (), and point multiplication () operations required by the protocol.$Hn_H En_{ENC} Mn_X Exp_{Exp} Pad_{ECP} Pmx_{ECX}$

5. ***Communication overhead:*** In Table 3.9, we compare the communication overhead of the proposed framework with four proposed protocols. The communication key that is being used to calculate the communication overhead is of size 32 bits to 64 bits. The hash key size is 32 bytes, and the compressed form of the RSA message is 24bits. The proposed framework decreases the computational overhead by 28%.

The proposed framework uses the current location of an IoT device within a designated area as the initial factor for authentication. To verify the location of an IoT device, the wireless channel characteristics of this IoT device are used.

Table 0.9 Comparison of Communication Overhead

Framework	**Messages**	**Exchanged Bytes**	**TOTAL bits**
[7]	4 + 4	268	2144
[35]	4 + 3	224	1792
[41]	2 + 2 + 1	163	1303
[42]	3 + 4	228	1824
[43]	7 + 6	368	2944
OURs	1 + 2	98	**784**

Moreover, the IoT device presenting unique features, known as the received signal power (RSSI) values, is used to compute the precise received power within the detectable distance without failure.

3.8 Conclusion

The current IoT ecosystem network structure is unable to provide a centralized security framework for every IoT device and application while protecting user data routing. Therefore, in this paper, we propose a security framework that will act as a heterogeneous security framework for every IoT device participating in the communication process. Indeed, we built the proposed framework on a novel SDN-based IoT payment framework architecture that decouples data and control planes of all IoT network devices. In the device's plane, the proposed framework separately allows the identification of IoT devices using RSSI power calculation for device authentication in simulated environments. In the data plane, the proposed framework provides a trusted data routing channel for IoT devices through settings in securing consumers' mobile payment applications against denial-of-service DDoS. As a result, the IoT network environment benefits from improved performance and management with secure user data relaying capabilities. In our current experimental settings, RSA public-private cryptography was deployed in the proposed framework to secure transmission links between IoT devices and controllers even under the presence of adversary IoT non-participating devices. Data routing and relaying capabilities are essential for the success of the operations and the processing in the consumer's centric applications, such as mobile payment applications. Moreover, the proposed framework adds more security features for existing consumer's centric applications, such as mobile payment applications in *IoT* and smart cities as presented in the broader design architecture. In future work, we will extend this work by using ECC (elliptic curve cryptographic) and evaluating the energy consumption of each node in a real non-simulated lab environment. However, this work can be applied to tackle more denial of services (DOS) attacks with AI-based learning optimization techniques and deployment techniques with nominal energy consumption once fully supported by the IoT ecosystem.

Acknowledgment

The authors wish to thank Dr. Franjieh El Khoury for her constructive comments and the proofreading of this paper.

CHAPTER 4 ARTICLE 2: IOT DEVICES MODULAR SECURITY APPROACH USING POSITIONING SECURITY ENGINE

published on July 8th, 2024 in IEEE Access by Jean Gerald Vincent Etibou and Samuel Pierre

Abstract— In this paper, we propose a modular security approach using a positioning security engine featuring Global Positioning System (GPS) location features that can uniquely identify the Internet of Things (IoT) user device. Our approach aims to reinforce the security and viability of IoT-centric solutions for various innovative applications, including IoT Mobile payment heterogeneous networks, communication services, safety, and location-based services integration. To achieve our goal of securitization and viability, we target consumer IoT devices equipped with built-in location-based GPS chips, which are vulnerable to hackers where the existing cryptographic authentication-based protocols demand power and computation resources required for authentication protocols is not sufficient to carry end to end secure transactions in an IoT environment. Therefore, to compensate for this lack of environment capability to carry the end-to-end secure transaction on IoT devices when emitting various radio signals, we implement a modular security approach to compensate for the lack of capabilities. This leads to an optimal security facilitated by Simple Public Key Infrastructure following the Pretty Good Privacy Web of Trust approach. Moreover, our implementation on the development board Arduino succeeded in providing an extended secure environment capable of carrying out secure transactions. The results show a communication success rate of 70, 80 and 90 percent between Security Engine component called modules, with 70 percent of successful Secure Sockets Layer (SSL) key exchange by every identified user in average 15 seconds simulation running time for every two by third round of simulation.

Index Terms—Authentication, identification, infrastructure, internet of things, location, modular security, simple public key infrastructure.

4.1 Introduction

The security and safety of emerging IoT devices as well as the expansion of variety of wireless connectivity connecting consumer IoT transmitting devices running purpose applications, such as mobile payment, are expected to reach more than 26 billion device numbers [76].

Wireless networks, especially ad hoc networks, benefit from device-to-device transmission signals to auto detect each IoT node device in a cooperative environment. This communication will be more heterogeneous and diverse in the context of IoT networks. Ad hoc IoT networks offer self-configuration and self-maintenance capabilities to IoT devices. Meanwhile such an environment with numerous IoT devices may pose a problem of identification of IoT transmitting devices to prevent fraud and phone cloning in case of RF cellular operators [77]-[78], and safety and security of very high frequency (VHF) radio networks, transmitter identification system [79]-[80]. Securing wireless channels to provide protected communication between IoT nodes devices in hostile environment is concerning and seemingly receiving tremendous attention from the research community requiring novel approaches and tools to increase security. Strong security has an impact on the success of application and services ranging from banking to business, finance, education, and industry [81].

The security of an IoT network can be enhanced if users can confirm their identity and if the RF (Radio Frequency) transmission of the IoT network devices is not deemed to be a threat. Thus, an attacker in an IoT network can spoof IoT network nodes and launch denial of service (DoD) attacks. Indeed, threats to the IoT network include man-in-the-middle attacks and reverse-engineering [82]. Although IoT networks benefit from accessibility, flexibility and usability while being exposed to privacy and controllability concern [83], IoT device identification is important to mitigate security problems in a large scale of IoT devices.

Therefore, in this paper, a modular security approach using positioning security engine is combined with GPS localization embedded in the security engine design. The modular security approach can be applied to IoT ad hoc networks security, spectrum resource management, wireless equipment safety certification, mobile phone network protection and more. In our modular security approach, we will include security over RF fingerprinting for enhancing IoT devices security and resolve the problem of IoT node device identification in a large scale IoT networks [84].

On one hand, combining geolocation and positioning details with RF details aims to enhance precision and accurate identification of the IoT RF transmitter. Since RF fingerprinting technique works on physical layer of wireless networks, RF fingerprints of IoT transmitter devices cannot be destroyed nor copied.

On the other hand, the fundamental of embedded security design is providing enhanced security in network inter communications, such as payment processing and other communications between IoT devices, while supporting a range of trust models. In instance, we propose to follow the Web of Trust (PGP) model [85] because of its large scale and decentralized architecture compatibility, which makes it relevant for IoT networks. In our approach, the secured engine design is motivated by providing a secure isolation environment in which secured and reliable transactions or communications are being carried between IoT device transmitters over offline connection in IoT wireless ad hoc network context [86].

The originality of this paper is that we consider offering a secure security engine to support IoT devices in IoT environment since IoT devices have limitation on processing resources to carry encryption, authentication, and identification processes. An IoT device can communicate to multiple gateways entities connecting the IoT device to the rest of the communications networks. Likewise, a modular security approach will offer security modules interface between the IoT device and the gateways. Therefore, we propose an IoT security modular system approach to enhance security for IoT devices and IoT network entities [83]. Our design of modular security offers a Device-Secured Module (DSM) to secure IoT devices using Enhanced Radio Frequency Fingerprinting (ERFF) by adding location awareness device information to overcome IoT devices' computational power limitations [99]. Then, we propose a Network Secured-Module (NSM) to secure IoT applications and network entities using insurance and trust logic methods applied to Simple Public Key Infrastructure (SPKI) for online and offline communications [87]. Finally, we simulate the system modularity functions using Arduino GSM / IoT board and obtain performance computational results that show effective IoT devices identification through DSM, applications, protocols, and network entities authentication through NSM.

The rest of the paper is organized as follows. Section 4.2 presents background and related work. Section 4.3 describes the proposed security engine system. Tests, results, and discussions are detailed in section 4.4. Finally, section 4.5 concludes this paper.

4.2 Background and Related Work

Most IoT receiver used in RF fingerprinting techniques are large and expensive [87]. However, new research has proven that RF fingerprinting techniques can be explored with low-cost affordable hardware, such as universal software radio peripheral receiver (USRP), compatible with IoT networks [88].

In network security, security keys must be properly issued and managed. However, large infrastructure like IoT networks makes it difficult for a universal security solution to be deployed. Moreover, IoT environments built on centralized security infrastructure are more vulnerable to threats when the central-key managing servers are attacked or compromised [89].

4.2.1 Background

In this section, we present the background work related to the design of our proposed system.

RF Fingerprinting for IoT Device

The first most known application of RF fingerprinting is the radar system and the tracking detection. In radar tracking systems, especially in military contexts, RF fingerprinting has been directed to identify a wide range of wireless communicating devices for authentication purposes. Similarly, IoT devices emitting Wi-Fi radio waveforms are exposing unique distinctive differences among the waveforms of different IoT radio devices in a large-scale IoT architecture [90].

Indeed, an RF fingerprinting system architecture consists of an acquisition sub-system, signal post-processing, feature extraction sub-system, dimensionality reduction sub-system and classifier sub-system [91]. The RF fingerprinting system architecture can be used to investigate RF hardware design imperfections. The imperfections of the electronic design of the IoT wireless radio may either contribute to differentiating between several devices or not [92].

Moreover, radio transmitters offer inherent nonlinearities that can be analyzed, then extracted as RF fingerprints of the signals [93].

In this paper, since the RF fingerprints are unique to the IoT transmitting device, we will combine the RF information into a secure engine part to generate security keys.

Security Engine (SE)

Secure engine (SE) communication makes use of Public Key Infrastructure (PKI) defines as an asymmetric key-based security infrastructure [94]. That security infrastructure needs a Certificate Authority (CA) as an organization that provides the security system with services for issuing and managing digital certificates. Moreover, the registration of the certificate is done by a Registration Authority (RA) acting as an enroller for the issued digital certificate. Therefore, a key that only a user has is called a private key (PvK), and a key that is open to any user else is called a public key (PbK). The use of PvK and PbK is explained as if the user owner of PvK encrypts communication using PbK, then the user can only decrypt the communication with PvK. In addition, if the user encrypts the communication using PvK, then the communication can be decrypted with the user who only owns PbK. However, in the offline and online mode of communication, successful implementation of PKI requires a Security engine that can handle both offline and online communication modes [95]. In this paper, we will propose an exchange key system that will limit the use of a centralized point of verification for resiliency and robustness following the Web of Trust (WoT) modeling approach.

Web of Trust (WoT)

The Web of Trust model is an alternative approach to the X.509 standard for building a Public Key Infrastructure solution. As the previous section mentioned, our system is designed as a Security Engine that relies on PKI, thus WoT approach will be recommended for our SE design.

Moreover, the mechanisms involved in WoT are decentralized. This allows every user of the system to sign another user’s public key(s) based on the experience with the parties. The security mechanism in WoT is based on credential verification, such as Pretty Good Privacy (PGP) [96] and GNU Privacy Guard (GnuPG) [97].

WoT is scalable and resilient as it does not suffer from a single point of failure. Thus, in this paper, our proposed system will offer resiliency and robustness.

Nevertheless, our SE must ensure that trusted users are allowed to exchange and sign keys, since there are many IoT device communications.

Insurance and Trust in Simple Public-Key Infrastructure (SPKI)

The insurance logic is described as a method for reasoning about how insured and signed keys may or not specifically derive statement about issuer and signer roles on those keys.

In the literature work, the insurance logic is an extension of the delegation logic of Lampson et al. [98] to strengthen authentication in a large-scale system such as distributed system. Therefore, this insurance logic method suits the IoT environment presented in this paper.

Insurance keys are assumed to be verified and easy to accomplish [86]. The user of the system is called an insurer and has a very important role. Keys are issued by users called issuers. These users assume that those keys are insured and known by themselves. In this paper, we do not require that all users must know all insurers' key. Rather, all insurers are credited with the ability to determine other insurer's keys. Therefore, insurers' keys should be very protected and remain accessible within insurers only. Notably, insurers that are deemed unscrupulous insurers, severely and too often misbehave. If detected by the Security Framework Bridge (SFB), those unscrupulous insurers must be deleted, banned, or punished.

Table 4.1 Abbreviations

SYMBOL	**MEANING**
$\Rightarrow$	"speaks"
$X \Rightarrow Y$	"*X* speaks *Y*" means *X* is public key owned by *Y*
X_Y	*X* is *Y's* public key
[$W,X,m]	key *X* is insured by *W* for up to *m* amount (insurance certificate)
$[\$W, X, m]_{X''}$	Insurance certificate signed with key *X'* (binding certificate)
$X_W\$_mW$	*W* signed public key with X_Wm amount (insurance certificate)
(A) → *(B)*	"*says*" relation between entities A and B
HE	Hybrid Element
NWE	Network Element
CSR	Recipient Secure Element
SSE	Sender Secure Element
PvK	Private Key
PbK	Public Key

Trust in insurance adds a trustworthy label to the entities involved in the transaction by assuming that the latter entity is trustworthy than a key signed by those same entities. Nonetheless, there is no absolute guarantee that those trustworthy entities could still be unscrupulous certificates' insurers. Table 4.1 illustrates the insurance logic annotation used in this paper.

4.2.2 Related Work

In this section, we present the related work in relation to the design of our proposed system.

RF Fingerprinting

Various proposed works from the literature present many different IoT security techniques. Most common techniques focus on hardware identification, such as the Network Interface Card (NIC) transmitting an IEEE 802.11 frame.

He et al. [99] discusses techniques related to RF fingerprinting for addressing challenges with localization-based approaches namely localization accuracy, network time delays, radio resources availability, signal level.

Numan et al. [100] proposed network interface card fine time measurement technique applied to machine learning method for mobile device indoor localization. The technique is limited to use of precise time measurements data as a main feature characteristic input to the machine learning model.

He et al. [101] proposed a technique based of time of arrival recorded on a cellular network for improving the accuracy localization of the user on the roam. The technique did not extend to heterogeneous networks like IoT networks.

Wu et al. [102] proposed a technique called PARADIS that collects presents hardware imperfections data information, then performs a machine learning based fingerprinting to identify the distinctive NIC. Nevertheless, this technique relies heavily on the performance of the chosen machine learning classification tool.

Based on protocols, Baldini et al. [103] proposed a method to fingerprint device based on common similar use transmission protocols given different devices transmitter. Their approach is based on the behavior of the devices for the observed same protocol, but it cannot be applied to IoT network since the IoT environment itself benefits from heterogeneous protocols, which will take longer and will consume a lot of resources to process all the data information for fingerprinting purposes only.

Based on network traffic analysis, Miettinen et al. [104] presented a fingerprinting technique for wireless devices by observing their emitting traffic on local area network (LAN). The method requires a dense traffic to capture network behavior to formulate signature for each device. But IoT network traffic is very minimal to generalize this technique to an entire IoT environment. In Addition, IoT networks have already a brownfield of legacy devices deployed and still active.

Therefore, a solution combining IoT based PKI and IoT fingerprinting will enhance security for IoT networks [105]. Radhakrishna et al. [106] proposed a mechanism based on location channel randomness pairing. The work has been tested only on their prototype and required implementation on all devices.

Public Key Infrastructure (PKI)

The PKI is the manager of the required key for both public users and private users. However, any user who intends to prove ownership of a key must hold a certificate verifiable by a Certificate Authority (CA).

The most common implementation of PKI is based on the X.509 standard [107] that verifies an entity's ownership of a CA's issued public key on the request. In this process, the verifying entity keeps the root certificate and trusts the CA if the certificate is successfully verified. Cooper et al. [108] have introduced PKI as a front-line security mechanism in the context of cryptography, where the communication and data security of the internet are threatened. X.509 based PKI standard research problems are as follows:

1- Lack of redundancy: Single point of failure for CA-based PKI.

2- Lack of traceability: CA-based PKI does not offer transparency.

3- Lack of recoverability: CA-based PKI must revoke certificates only option when found CA compromised by rogue attack.

4.3 Proposed System

The proposed security system combines Enhanced RF fingerprinting (ERFF) with device location information with Public Key Infrastructure (PKI) for securing IoT wireless communications. The

Enhanced RF fingerprinting by device location module named DSM – Device Secured Module - offers protection through device identification for IoT hardware and radio communications, while the NSM – Network Secured Module - offers protection through service authentication for protocols and applications exchanges. Both NSM and SDM are designed to be embedded into a Security Engine (SE).

4.3.1 System Overview

The proposed system architecture, as shown in Figure 4.1, is composed of two main blocks: Block 1 is Device-Secured Module (DSM) and block 2 is Network-Secured Module (NSM).

According to the system architecture design, an identification request to DSM by an IoT device before any user interface transaction is to be permitted. Then, this IoT device is prepared to forward the control to the user interface application assuming all communications channels are established.

The IoT device receives the previous Public Identification Key (PbIDK) from the user through User Interface (UI) application, and the IoT device checks its status of identification permission to allow the user to gain access to UI's application requested services.

DSM Module

The main objective of the DSM module is to build up an effective and secured IoT device environment through device identification mechanisms such as RF Fingerprinting [115]. Moreover, DSM offers the first stage of protection through RF fingerprinting identification depending on the IoT environment and the service requested by the application of the user interface (UI). DSM is basically in charge of data acquisition, feature extraction and classification.

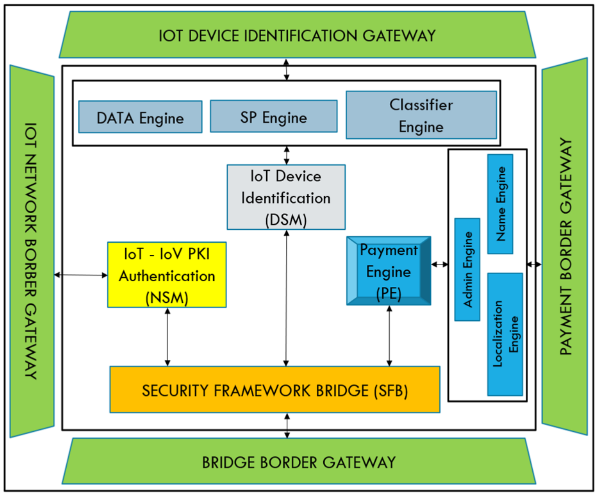

IOT DEVICE IDENTIFICATION GATEWAY
DATA Engine
SP Engine
Classifier Engine
IoT Device Identification (DSM)
IoT - IoV PKI Authentication (NSM)
Payment Engine (PE)
Admin Engine
Name Engine
Localization Engine
SECURITY FRAMEWORK BRIDGE (SFB)
IOT NETWORK BORBER GATEWAY
PAYMENT BORDER GATEWAY
BRIDGE BORDER GATEWAY

Figure 4.1 Proposed Security Engine Infrastructure SE.

Data acquisition is performed by an acquisition submodule, which acquires and digitizes radio signals from connected IoT wireless devices [116]. DSM performs data acquisition in either active or passive mode [117].

RF features extraction [118] is the next process for RF Fingerprinting, which generates characteristic attributes from the raw signal emitted by the IoT devices. This activity of features extraction is governed by any hypothetical extraction concept that minimizes the input dimension to achieve the efficiency of the extraction process. Therefore, the length of the feature vector will be reduced without missing elements needed to perform the next step with the classification process.

The best description of classification [119] is the process initiated to perform a task on a trained network subjected to respond when an input vector like a learned vector is presented. The literature indicates that most classifiers follow an approach initially called Bayesian [120] classification to provide a common solution to pattern classification problems. In our proposed SE, the DSM achieves the protection level 1 of the IoT networks and environments, as shown in Figure 4.2.

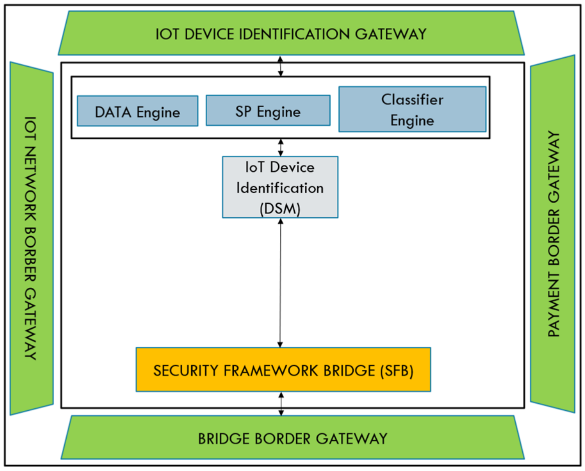

IOT DEVICE IDENTIFICATION GATEWAY
IOT NETWORK BORBER GATEWAY
DATA Engine
SP Engine
Classifier Engine
IoT Device Identification (DSM)
SECURITY FRAMEWORK BRIDGE (SFB)
PAYMENT BORDER GATEWAY
BRIDGE BORDER GATEWAY

Figure 4.2 Device security Module (DSM) system module of SE.

NSM Module

The NSM is deployed by DSM to validate the claim of device identification before proceeding to payment realization for instance. The IoT device gets its Public Key Identification PbKID from User Interface (UI). UI checks the validity of his permission request for the certificate. The validation process's next step involves the Security Framework Bridge (SFB).

The NSM connects to the Payment Engine (PE) through SFB for payment proceedings and realization.

The NSM responds to DSM through SFB to complete validating claims of IoT devices successfully compliant to protection level 1. NSM acts as a protection level 2 in our proposed SE system, as shown in Figure 4.3.

Security Framework Bridge (SFB)

The Security Framework Bridge (SFB) functions as Trust Third Party (TTP). IoT devices request a digital certificate through forwarded request by NSM. SFB includes two on demand functions that are essential for the integrity of the SE system. These functions are integrated into the SFB modules as rollback and buffer.

Rollback is essential to the PE transaction recovery if there is any issue with the payment type's transaction completion issues.

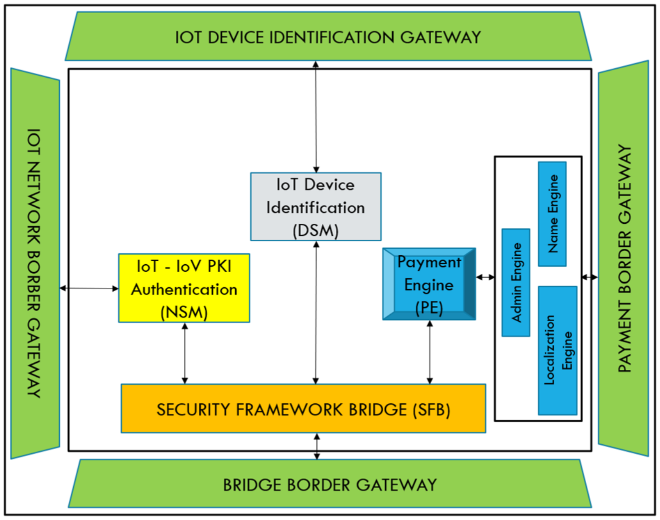

IOT DEVICE IDENTIFICATION GATEWAY
IOT NETWORK BORBER GATEWAY
PAYMENT BORDER GATEWAY
IoT Device Identification (DSM)
IoT - IoV PKI Authentication (NSM)
Payment Engine (PE)
Admin Engine
Name Engine
Localization Engine
SECURITY FRAMEWORK BRIDGE (SFB)
BRIDGE BORDER GATEWAY

Figure 4.3 Network Security Module (NSM) system module of SE.

Buffer is offering the property of a caching transactions for the SFB as a second volatile temporary read only memory. It can be used to speed up the processing time for the transactions and provide the additional resources that might be needed in case a transaction log exceeds the original length and size.

SFB confirms DSM protection level 1 status (i.e., success or failure). Then, after following its algorithm, SFB confirms NSM deployment and protection phase 2 status (i.e., success or failure). The next task for SFB is to send session keys to UI for decryption. In our proposed SE system, SFB acts as a bridge to DSM, NSM, PE and UI, as shown in Figure 4.4.

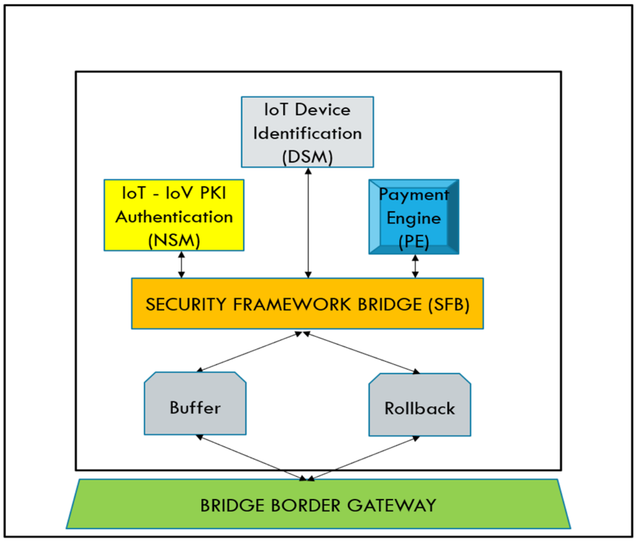

IoT Device Identification (DSM)
IoT - IoV PKI Authentication (NSM)
Payment Engine (PE)
SECURITY FRAMEWORK BRIDGE (SFB)
Buffer
Rollback
BRIDGE BORDER GATEWAY

Figure 4.4 Security Framework Bridge (SFB) system module of SE.

4.3.2 System Workflows

Certificates and Key Exchange Scheme

The challenge in this large-scale system is the certificate exchange to distribute to many IoT devices following the PKI infrastructure. The exchange scheme is shown in Figure 4.5 followed by the exchange mechanisms. In the key exchange system, the IoT device makes a request to the NSM module through the IoT network border gateway. This latter can be connected to any third-party PKI distribution infrastructure such as a public key distribution system in blockchain [121]. It is worth noting that the NSM module in Figure 4.3 represents the abstract of the SE system. The public key for IoT devices stores additional information about the device ID, OS version and Processing Unit (PU) capabilities. Moreover, one-time communication session key's property can reduce the risk of key's leakage such as reverse engineering key retrieval, since the session key is completely updated from IoT device buffer memory after the set timeout has expired.

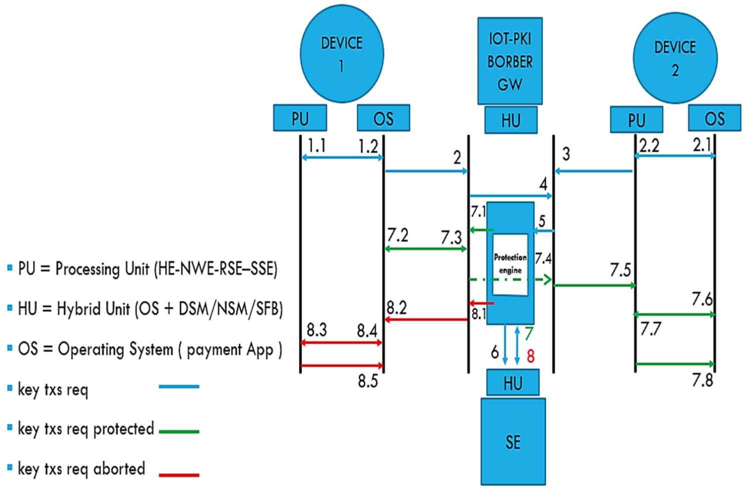


Figure 4.5 Proposed Certificates and Keys exchange scheme.

(1) Start
(1.1) Device 1 requests UI credentials (1.2)
(2.1) Device 2 requests UI credentials (2.2)
(2) Device 1 Application Interface adds UI credentials to HU/DSM
(3) Device 2 Application Interface adds UI credentials to HU/DSM
(4) HU/DSM requests public key of device 2
(5) Protection Engine receives public key of device 2
(6) HU/NSM validates public key
(7) HU/SFB authorizes / aborts (8) sessions between device 1 and device 2.
(7.1) Confirmation session authorization to device 1
(7.2) Device 1 generates session key
(7.3) HU/SFB encrypts the session using device 2 public key
(7.4) Confirmation session authorization to device 2
(7.5) PU/Device 2 pulls device 1 public key information
(7.6) Device 2 application interface receives session key to decrypts
(7.7) Device 2 decrypts the session key with its private key
(7.8) Service request transferred to border gateway
(8.1) Confirmation session abortion to device 1
(8.2) Device 1 generates rollback session key
(8.3) PU/Device 1 revokes device 2 public key information
(8.4) Device 1 application interface release session

(8.5) OS/Application notifier update
(9) Stop.

Insurance and Trust Scheme

Insurance and trust scheme refers to authentication in large-scale distributed systems according to LABW logic [98], interprets a certificate as a statement representation. There are two paths to building insurance: the shorter path and the longer path.

- **Shortest path**

The shortest path in insurance logic works without trust with the other users of the system. It is described as follows:

(1) AND $[D_1, K_{D_1}]_{K_{D_2}} [\$D_3, K_{D_2}]_{K_{D_3}}$
(2) $K_{D_3} \Rightarrow D_3$
(1) is interpreted as
(3) K_{D_2} says $K_{D_1} \Rightarrow$ AND says $\$D_1 K_{D_3} K_{D_2} D_3$
(1) AND (3) give
(4) says $D_3 K_{D_2} \$D_3$
(5) $K_{D_2} \$D_3$
(2) AND (5) give
(6) $(K_{D_1} \Rightarrow)\$D_1 D_3$

With the use of the shorter path, we can conclude that link relation between and $.D_3$ has insured the $D_1 K_{D_1}$

Moreover, because (6) may not apply since:

(7) $K_{D_1} \Rightarrow$ is true.D_1

- **Longer path**

The longer path in insurance logic combines trust and insurance in the same path. It is described as follows:

(1) $[D_4, K_{D_4}]_{K_{D_1}}$
(2) if D_1 says $(K_{D_4} \Rightarrow)$ then D_4 $K_{D_4} \Rightarrow D_4$
(3) K_{D_1} says $(K_{D_4} \Rightarrow D_4)$
(4) $(K_{D_1} \Rightarrow D_1) D_3$, OR $D_3\$$
(4) is true
(5) $K_{D_4} \Rightarrow D_4$

If is damaged by any user device, thenK_{D_4}

(5) is false

Trust returns to since it was the first trustworthy introducer with no ever damaged or reported compromised.$D_1 K_{D_1}$

Security Engine’s Proposed Algorithm

Algorithm 1 Security Engine Core Algorithm

```
Input: S(V).
V={M∪K},
M={DSM,NSM,SFB,PE},
K_(n,m)={PvIDK1,PbIDK2,PvIDKn....PbIDKm}.
Output: Protection phase state S.

1 : DSM protection phase 1 STARTS
1.1 : If device's  RF FRIGERPRINTING passed then DSM requests confirmation of Native GPS service resource availability,
1.2 : DSM updates  latest device's location with geohash code precision t (seconds taken to update gps information)
1.2.1 : DSM requests SFB to deploy NSM
1.3 : else DSM requests SFB protection phase 1 at p (time taken to fail phase 1) to be aborted.
2:  NSM protection phase 2 STARTS
2.1: if NSM validates Device's ID claims by ISSUING SIGNED ID CERTIFICATE using USER INTERFACE PbIDK then NSM replies to DSM copy of USER INTERFACE PbIDK for user payment application processing through PE
2.2 : SFB sends session KEY to USER INTERFACE with SIGNED CERTIFICATE through PE for user to decrypt with PvIDK
2.3 : else NSM requests SFB protection phase 2 at p' (time taken to fail phase 2) to be aborted.
3: SFB confirms Protection Phase 1 AND Protection Phase 2  SUCCESS then forwards ACK to PE to proceed with user payment transaction
3.1: if user failed decryption with PvIDK then PE request SFB to abort payment or service request
4: else if SFB returns NSM Protection Phase 1 status and DSM Protection Phase 2 status then update PE to proceed with offline user payment or service request.
```

The algorithm presents the design of the Security Engine (SE) representing the virtual connection of an IoT secure environment connecting to four gateways (i.e., the IoT device gateway, the IoT network border gateway, the payment boarder gateway, and the bridge border gateway). The algorithm I for the SE has a beginning phase with the module in charge of the device security level described as level 1 and named DSM. DSM protection phase 1 starts when the IoT device starts transmitting radio signals. The device's RF fingerprinting details need to pass the stage of full transmission for the DSM to make a request to the next level module using the positioning details. DSM requests confirmation of GPS service resource availability and demands SFB to deploy NSM.

The SFB routes the requests made from DSM to NSM and vice versa. Essentially, NSM oversees the protection phase 2. Protection phase 2 starts with process validation and PKI keys generation and exchange processes.

NSM validates IoT device's identification claims by issuing signed certificate using user interface PbIDK, then NSM replies to DSM with a copy of user interface PbIDK for user payment application processing through Payment Engine (PE).

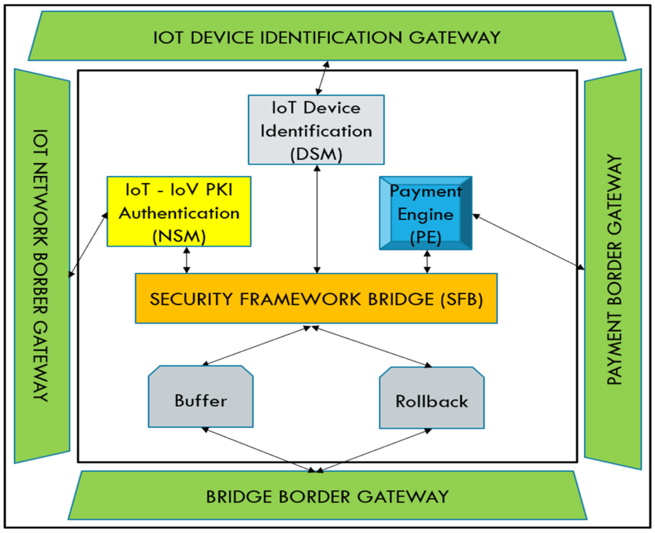

IOT DEVICE IDENTIFICATION GATEWAY
IOT NETWORK BORBER GATEWAY
PAYMENT BORDER GATEWAY
IoT Device Identification (DSM)
IoT - IoV PKI Authentication (NSM)
Payment Engine (PE)
SECURITY FRAMEWORK BRIDGE (SFB)
Buffer
Rollback
BRIDGE BORDER GATEWAY

Figure 4.6 Security Framework Bridge (SFB) system with payment protection.

The overall session is monitored by the SFB. This latter sends session key to user interface with signed certificate through PE for user to decrypt with PvIDK.

SFB confirms whether protection phase 1 and protection phase 2 succeeded, then forwards ACK message to PE to proceed with user payment transaction, as shown in Figure 4.6 with the SE core payment modules components.

Security Engine's Proposed Modelling Block Diagram

Signal generation can be done using GNU radio SDR software for recording IoT device transmitted raw signal at any different dedicated use mode of operation, such as during phone calls, mobile date payment, and sending MMS and SMS, as shown in Figure 4.7.

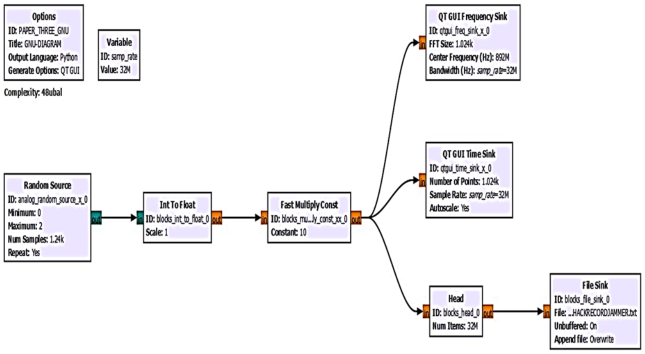


Figure 4.7 Proposed SE GNU representation block diagram.

Table 4.2 MKR 1400 Specification

Device Model	**32-bit development board**				
connectivity	2G / 3G				
Chipset	ATSAMD21				
Clock	48 Mhz				
Memory	256 KB FLASH		32 KB SRAM		
interfaceS	USB	SPI	12C	12S	UART
VOLTAGES	5V INPUT		3.3V OPERATING		
Pinout	22 INPUT		12 PWN	7/1 ANALOG	
Dimensions	67.64 x 25 mm				

Each IoT device can operate between a range (890 MHz to 3500 MHz) of LTE / 5G frequency of the carrier cellular mobile network following orthogonal frequency-division multiplexing (OFDM) propagation. The RF specifications make the design of our proposed Security Engine compatible with the arduino test experimental board GSM MKR [125], as illustrated in Table 4.2.

4.4 Results and discussions

4.4.1 Experimental setup

Table 4.3 Experimental Setup

item	**description**
MKR 1400	32-bit development testing board
Antenna	Dipole Pentaband Waterproof Antenna GSM 850 / 900 / 1800 / 1900 MHz and UMTS bands
SIM Card	Arduino SIM
IDE	ARDUINO IDE 1.8.19
PC	Windows 10 pro OS, 8 GB, 512 GB, core i5 vPro

For this experiment, Table 4.3 presents our experimental setup. Our goals are to obtain significant results, and we assume signal degradation reduction due to noise is performed at the data acquisition stage using high-quality signal measurement [126]. For the benefits of the simulation, we use Software Defined Radio (SDR) software GNU Radio [127] as a signal processing engine while the device hardware provides the RF front end. GNU Radio provides an extensive library of processing blocks running functions and algorithms, such as encoding, decoding, mixing, filtering, equalizing and packet handling. GNU Radio modeling is tested by an IoT GSM Arduino board for running the SE algorithm sub-routine in passive mode, since we interact with the system by SMS messaging.

Successively users will send messages to the SE system through the MKR board processing user identification.

The passive mode of data acquisition is used in our experimental setup since Global System for Mobile Communication (GSM) standard group of previous works had applied passive signal of data acquisition for device identification purposes for cellular telephones communicating with real or emulated base node stations [125].

GSM tests are conducted using the arduino board MKR 1400. The MKR board has an inbuilt GSM modem, and we attached a dedicated GPS module for computing the GPS coordinated positioning metrics as longitude, latitude, and accuracy [126].

The MKR GSM 1400 is a great option for GSM connectivity development. The MKR board is using the popular Arm Cortex-M0 32-bit SAMD21 processor. The development board also features the powerful u-blox SARA-U201 module and the ECC508 crypto-chip for security.

We assume that in our setup individual users through the user interfaces, make their own unsupported decisions as to the thrusted-ness of the certificates' introducers.

Instead, we use the internal MKR SSL management to comply with the MKR board resource constraint.

Table 4.4 Iot Device Interaction

Test #	Round	Status	DSM Phase 1	NSM Phase 2	SFB	SE	Timestamp
1	3	success	3	3	3	6	20:03:09.048
		fail	0	0	0	0	-
2	5	success	4	4	4	12	20:03:09.533
		fail	1	1	1	3	20:03:21.487
3	10	success	8	7	7	22	20:04:43.725
		fail	2	3	3	8	20:08:15.197

In our current experimental setup, we made the abstraction of possible generation of audited key [127] – [128] to privilege faster computation and keep this latter within the circumscription of the proposed Security Engine (SE).

4.4.2 Results

In the subsequent tables' result, test 1 shows 3 attempts from user toward the SE, test 2 shows 5 attempts and test 3 shows 10 attempts respectively. The

Tableau 4.5 SSL Clients and Key Exchange

Test #	Round	Status	DSM Phase 1	NSM Phase 2	SFB	SE	Timestamp
1	3	success	2	2	2	6	19:21:03.608 -> AT
		fail	1	1	1	3	19:21:14.982 -> ERROR
2	5	success	4	4	4	12	19:21:15.088 -> OK
		fail	1	1	1	3	19:21:15.088 -> ERROR
3	10	success	7	8	7	22	19:21:15.412 -> OK
		fail	3	2	3	7	19:21:16.580 -> ERROR

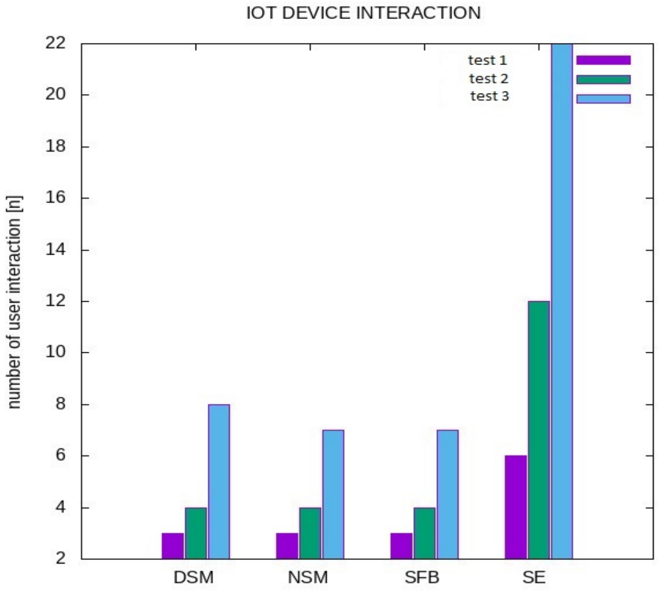


Figure 4.8 Number of interactions with SE

timestamp shows the recorded time duration of the test when the security modules DSM, NSM and SFB successfully respond to the user attempts respectively.

Table 4.4 illustrates IoT device interaction with IoT user application and device authentication results.

Table 4.5 represents PKI key exchange results.

Table 4.6 represents the performance metrics for the proposed SE system.

4.4.3 Discussion of results

Our SE's features, tested on the arduino MKR 1400, are compared against IoT 1 and IoT 2 devices in Table 4.7. Unlike IoT 1 and IoT 2, our SE does not require an additional device companion to be able to offer user device interaction and SSL authentication. In Table 4.6, DSM phase 1 for instance, the subtraction between the start timestamp and the stop timestamp reveals that phase 1 at test 1 elapsed around 43 seconds, 15 seconds at test 2, 13 seconds at test 3 respectively while recorded temperature of the MKR board range from 31.18 degrees Celsius at test 1, to 32.15 degrees at test 2, to

Table 4.6 SE Performance Metrics

Test #	Round	Simulation		DSM Phase 1	NSM Phase 2	SFB	SE
1	3	timestamp	start	20:02:26.344 –	20:03:09.533 -	20:03:45.544 -	20:03:45.546 -
			stop	20:03:09.048	20:03:45.016	20:03:45.545	20:03:46.540
		temperature		31.18	31.72	31.61	31.61
2	5	timestamp	start	20:04:29.658 -	20:04:44.269 -	20:04:57.376 -	20:04:57.378 -
			stop	20:04:43.725	20:04:56.827	20:04:57.377	20:04:58.402
		temperature		32.15	32.37	32.58	32.58
3	10	timestamp	start	20:06:17.246 -	20:06:30.628 -	20:07:17.174 -	20:07:18.751 -
			stop	20:06:30.113	20:06:42.993	20:07:17.718	20:08:02.080
		temperature		32.90	33.23	33.12	33.12

32.90 degrees at test 3. Since at phase 1, DSM is responsible for user authentication after successful device interactions, the average time is around 15 seconds to be consistent with the timestamp recorded in Table 4.4 and Table 4.5.

Our approach to device identification makes use of the GSM module built into the MKR board. This latter uses the GSM library to send and receive SMS in an effective way. Moreover, the GSM library allows SE system to connect to internet through the GPRS networks by using web clients for the requester (SMS sender) and server for the responder (SMS receiver). In addition, the uniqueness of the user mobile number helps to bind the user application requests with the api level capability of the requested services, as shown in Figure 4.8.

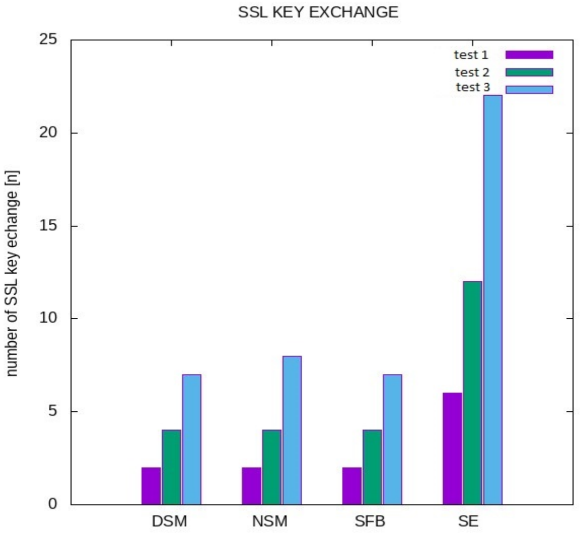


Figure 4.9 Number of SSL key exchange with SE.

Our approach to authenticate the users using SSL management of the MKR board brings SSL clients in the based class for all GPRS

SSL client-based calls. The SSL is not called directly but invoked whenever the system uses a function withing the MQTT broker client that relies on it. MQTT client library implementation is ported to support ESP32/S2/S3/C3, WT32_ETH01 (ESP32 + LAN8720), ESP32 using LwIP ENC28J60, W5500, W6100 or LAN8720. The same library supports TLS/SSL for MQTTS clients, which is an ideal choice for our proposed SE and makes it compatible with most communication protocols running on IoT devices. The proposed SE system allows creating a client that always connects SSL to the specified IP address using the GPRS mobile operator sockets and port, even if client.connect() is used instead of client.connectSSL(). Our implementation helps us to find that it is useful even if we have a library that accepts only plain clients, but we want to force the client to use SSL, keeping the same method names as the non-SSL client. This makes our proposed SE system secure for any user client IoT connected devices or applications, as shown in Figure 4.9.

Table 4.7 IoT Devices Comparison

Device Model	**IoT 1**	**IoT 2**	**Our SE MKR 1400**
Platform	ArmV8	32 bit ARM	ATSAMD21 32 bit
Open-Source model	no	no	yes
On device user SmS	optional	optional	yes
On device SSL	no	no	yes
External encryption resource	required	required	no
IoT Devices interaction	yes	yes	yes
Mobile Device companion	required	required	no
GPS Dual frequency	yes	yes	yes

Our approach based on Public Key Infrastructure in the large-scale environment of devices like IoT, is useful if the certificates are self-signed by users called introducers. There are no needs for a third trust party (TTP) such as certification authorities in the current form of our proposed approach. Users of the proposed system can decide who are trustworthy certificates' introducers or who is not. Our system does not follow any hierarchy path of insuring the certificates' introducer. In such a case, some users will or will not certify other's keys and in return, such users may or may not have their own keys insured by others.

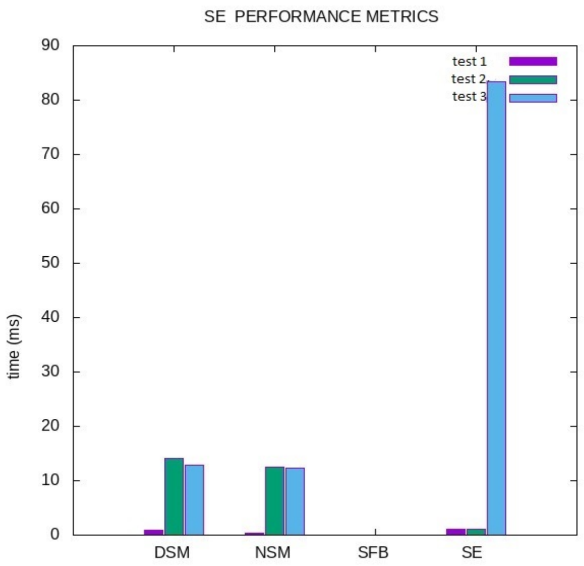


Figure 4.10 SE response time performance metrics.

Figure 4.10 depicts the overall time, and the performance of the proposed SE system taken into completing phase 1 and phase 2 till the final cycle loop. We noticed that SFB was fast enough to forward the final request as the SE overall time is lesser in test 1 and test 2.

4.5 Conclusion

In this paper, we presented a new approach towards enhancing IoT security relying on secure transmission in a secure environment for IoT devices with specific identification. For any application to IoT, the security of the transaction is governed by the SPKI within a proposed secure engine infrastructure that prevents dual expenditure in offline communication while showing 70, 80 and 90 percent communication success between DSM, NSM and SFB with 70 percent of successful SSL secure key exchanges by every identified user. Our research work can be recommended for secure anonymity, privacy, and non-repudiation on user application purposes using a registered IoT device under providing a set of features where the IoT device benefits from using the proposed SE for additional computational resources for optimal performance and integrity of the transmitted data.

As future works, some improvement needs to be addressed in terms of computational time and responsiveness of our SE system since the testing development Arduino board perhaps while operating had shown some delays connecting with using GSM network. Furthermore, we will try to include more options to the current design of the SE to target large-scale environments, such as IoT cloud, crypto exchanges and blockchain with limitations on artificial intelligence (AI) models since our current system does not include yet an energy-efficiency model for AI computations.

Acknowledgment

The authors wish to thank Dr. Franjieh El Khoury for her constructive comments and the proofreading of this paper.

CHAPTER 5 ARTICLE 3: MODULAR SECURITY ENGINE CYBER-PHYSICAL RESILIENCY APPROACH USING DIGITAL REPLICATION

published on September 30th, 2025 in IEEE Access by Jean Gerald Vincent Etibou and Samuel Pierre

***Abstract*—**In this paper, we propose an IoT device cyber physical resiliency approach using a cross-domain device replication that can replicate and securely transfer the profile of the Internet of Things (IoT) user device to another. Our approach aims to offer a cybersecurity abstraction layer to offer continuation of the security engine service when the IoT user device is facing interruption of services in case of IoT-centric solutions, including IoT Mobile payment networks communication services, safety communications with integrated location-based services. In a world of cybersecurity where any IoT user devices are vulnerable to hackers, existing IoT devices require them to remain efficiently powered, connected to carry end-to-end secure transaction in an IoT environment without failure. To achieve our goal of resilience and recoverability, we upgraded the IoT modular security engine features with digital replication functions and secure profile transfer capabilities to effectively complete started transactions in situations where lacking sufficient power to transmit or in case of an abrupt connection loss due to network handovers. This leads to integration of IoT cross-environment optimal security enhancements into the existing Simple Public Key Infrastructure following the Pretty Good Privacy Web of Trust approach with Secure Device Profile and Data Processing. Moreover, our implementation succeeded in providing a service continuation on the replicated IoT device, preserving and offering a resilient secure capable environment for carrying secure transactions. The results show a communication success rate of 90 percent between all Security Engine components (DSM, CNSM, SFB) called modules on replicated IoT devices while improving the simulation running time to run two times longer (30 seconds) in each round of tests with 10 per cent faster response and recovery time of 7 seconds.

***Index Terms*—** Authentication, cybersecurity, identification, infrastructure, internet of things, location, modular security, resiliency.

5.1 Introduction

The security and safety of emerging IoT devices as well as the expansion of variety of wireless connectivity connecting consumer IoT transmitting devices running purpose applications, such as mobile payment, are expected to reach more than 26 billion device numbers [129].

In the growing world of connectivity focused users, many connected devices share the same networks and connection resources. The connected devices continuously self-detect, connect, send and receive data information and more to other devices operating the same over the networks exposing them to serious cybersecurity threats due to the possibility of spoofing over the networks lacking authentication or encryption [130].

Wireless networks, especially ad hoc networks, benefit from device-to-device transmission signals to auto detect each IoT node devices in a cooperative environment. This communication will be more heterogeneous and diverse in the context of IoT networks. Ad hoc IoT networks offer self-configuration and self-maintenance capabilities to IoT devices. Meanwhile such an environment with numerous IoT devices may pose a problem of identification of IoT transmitting devices to prevent fraud and phone cloning in case of RF cellular operators [131]-[132], and safety and security of very high frequency (VHF) radio networks, transmitter identification system [133]-[134]. Securing wireless channels to provide protected communication between IoT nodes devices in hostile environment is concerning and seemingly receiving tremendous attention from the research community requiring novel approaches and tools to increase security. Strong security has an impact on the success of application and services ranging from banking to business, finance, education, and industry [135].

The security of an IoT network can be enhanced if users can confirm their identity and if the RF (Radio Frequency) transmission of the IoT network devices is not deemed to be a threat. Thus, an attacker in an IoT network can spoof IoT network nodes and launch denial of service (DoD) attacks.

Indeed, threats to the IoT network include man-in-the-middle attacks and reverse-engineering [136]. Although IoT networks benefit from accessibility, flexibility and usability while being exposed to privacy and controllability concern [137], IoT device identification is important to mitigate security problems on a large scale of IoT devices.

At large scale, an IoT device as a sub cyber-physical system is embedded with multiple sub-system modules (i.e., standalone booting operating system, energy, networking) with heterogenous goals and complexity of cross environment interactions [138].

Therefore, in this paper, a cyber-physical resiliency approach using a cross-domain device replication is embedded in the security engine upgraded design. The cyber-physical resiliency approach can be applied to cross IoT environment ad-hoc networking system, wireless equipment safety certification, cellular network virtual radio access protection and more. Our cyber-physical resiliency approach will enhance security over native RF fingerprinting for enhancing IoT devices security and resolve the problem of IoT node device identification in large scale IoT networks [139].

On one hand, the IoT device will communicate through an identification system that periodically gathers the device information data, type and location into an IoT unique device profile [140]. The system is combining geolocation and positioning details with RF details aims to enhance precision and accurate identification of the IoT RF transmitter. Since RF fingerprinting technique works on physical layer of wireless networks, RF fingerprints of IoT transmitter devices cannot be destroyed nor copied.

On the other hand, there will be an amount of data to deliver to IoT destination endpoint device. We propose a path-based data transfer mechanism injecting unique identifier packet and then sequentially rerouting them to IoT potential destination endpoints [141]. The fundamental of embedded security design is providing enhanced security in network inter communications, such as payment processing and other communications between IoT devices, while supporting a range of trust models. In instance, we follow the Web of Trust (PGP) model [142] because of its large scale and decentralized architecture compatibility, which makes it relevant for IoT networks. In our approach, the upgraded secured engine design is motivated by providing a secure cross environment in which secured, reliable and recoverable transactions or communications are carried between IoT device transmitters over offline connection in cross environment IoT wireless ad-hoc network context [143].

The originality of this paper is that we consider offering a cyber-attack resilient security engine to support IoT devices in cross IoT environment to continue encryption, authentication, and identification processes even in situations of lacking sufficient power to transmit or in case of an abrupt connection loss due to network handovers.

An IoT device can communicate to multiple gateways entities connecting the IoT device to the rest of the communications networks. Likewise, a modular security approach will offer security modules interface between the IoT device and the cross-environment gateways. Therefore, we propose an IoT device cyber physical resiliency system approach to offer a cybersecurity abstraction layer to offer continuation of the security engine service when the IoT user device is facing interruption of services on cross-environment IoT network entities [137]. Our upgraded design main modules consist of a Device-Secured Module (DSM) to secure IoT devices using Enhanced Radio Frequency Fingerprinting (ERFF) [163]. Then, we propose a Cross Network Secured-Module (CNSM) to secure IoT applications and cross environment network entities using insurance and trust logic methods applied to Simple Public Key Infrastructure (SPKI) for online and offline communications [144] with The Security Framework Bridge (SFB) functioning as Trust Third Party (TTP) for preserving the integrity of the Security Engine system. Finally, we simulate the system modularity functions using Arduino IoT board and obtain performance computational results that show a communication success between all Security Engine modules on both Physical IoT devices replicated IoT devices through DSM, applications, protocols, and cross environment network entities authentication through CNSM.

The rest of the paper is organized as follows. Section 5.2 presents background and related work. Section 5.3 describes the upgraded security engine system. Tests, results, and discussions are detailed in section 5.4. Finally, section 5.5 concludes this paper.

5.2 Background and Related Work

In the cybersecurity world, cyber-physical systems need to encompass resilience integrating data computing and sensors data gathering strategies while withstanding unexpected events like networks interruptions and loss of power [145].

Most IoT receivers used in RF fingerprinting techniques are large and expensive [144]. However, new research has proven that RF fingerprinting techniques can be explored with low-cost affordable hardware, such as universal software radio peripheral receiver (USRP), compatible with IoT networks [146].

In network security, security keys must be properly issued and managed. However, large infrastructure like cross IoT networks makes it difficult for a universal security solution to be deployed and managed. Moreover, cross IoT environments built on centralized security infrastructure are more vulnerable to cybersecurity threats when the centralized key managing servers are attacked or compromised [147].

5.2.1 Background

Resilient IoT Device

IoT devices are built with smart sensors which communicate over cross network environment as they gather physical data to be used by cyber existing components [148]. In contrast to traditional systems, IoT devices built in smart sensing hardware gain broader insight into control of efficiency, energy consumption, redundancy, and resiliency to any threat to completion of transactions [149].

The type of resiliency for IoT device is determinate by the approach, the design and the strategy. In cross IoT environments, the paradigm of computing at the edge, offers more resilient control over cyber – physical systems. The sensors inside the cyber physical system can independently perform their functions and analytics bringing the system to hinder operations while preserving critical functionality of the core of the system [150].

RF Fingerprinting for IoT Device

The first most known application of RF fingerprinting is the radar system and the tracking detection. In radar tracking systems, especially in military contexts, RF fingerprinting has been directed to identify a wide range of wireless communicating devices for authentication purposes. Similarly, IoT devices emitting Wi-Fi radio waveforms are exposing unique distinctive differences among the waveforms of different IoT radio devices in a large-scale IoT architecture [151].

Indeed, an RF fingerprinting system architecture consists of an acquisition sub-system, signal post-processing, feature extraction sub-system, dimensionality reduction sub-system and classifier sub-system [152]. The RF fingerprinting system architecture can be used to investigate RF hardware design imperfections. The imperfections of the electronic design of the IoT wireless radio may either contribute to differentiating between several devices or not [153].

Moreover, radio transmitters offer inherent nonlinearities that can be analyzed, then extracted as RF fingerprints of the signals [154].

In this paper, since the RF fingerprints are unique to the IoT transmitting device, we will combine the RF information into a secure engine part to generate security keys.

Security Engine (SE)

Secure engine (SE) communication makes use of Public Key Infrastructure (PKI) defines as an asymmetric key-based security infrastructure [155]. That security infrastructure needs a Certificate Authority (CA) as an organization that provides the security system with services for issuing and managing digital certificates. Moreover, the registration of the certificate is done by a Registration Authority (RA) acting as an enroller for the issued digital certificate. Therefore, a key that only a user has is called a private key (PvK), and a key that is open to any user else is called a public key (PbK). The use of PvK and PbK is explained as if the user owner of PvK encrypts communication using PbK, then the user can only decrypt the communication with PvK. In addition, if the user encrypts the communication using PvK, then the communication can be decrypted with the user who only respective PbK. However, in the offline and online mode of communication, successful implementation of PKI requires a Security engine that can handle both offline and online communication modes [156]. In this paper, we will propose an exchange key system that will limit the use of a centralized point of verification for resiliency and robustness following the Web of Trust (WoT) modeling approach.

Web of Trust (WoT)

The Web of Trust model is an alternative approach to the X.509 standard for building a Public Key Infrastructure solution. As the previous section mentioned, our system is designed as a Security Engine that relies on PKI, thus WoT approach will be recommended for our SE design.

Moreover, the mechanisms involved in WoT are decentralized. This allows every user of the system to sign another user’s public key(s) based on the experience with the parties. The security mechanism in WoT is based on credential verification, such as Pretty Good Privacy (PGP) [157] and GNU Privacy Guard (GnuPG) [158].

WoT is scalable and resilient as it does not suffer from a single point of failure. Thus, in this paper, our proposed system will offer resiliency and robustness.

Nevertheless, our upgraded design of SE ensures that trusted users are allowed to exchange and sign keys, since there are many IoT device communications.

Insurance and Trust in Simple Public-key Infrastructure (SPKI)

The insurance logic is described as a method for reasoning about how insured and signed keys may or not specifically derive statement about issuer and signer roles on those keys.

Table 0.1 Abbreviations

SYMBOL	**MEANING**
$\Rightarrow$	“speaks”
$X \Rightarrow Y$	“*X* speaks *Y*” means *X* is public key owned by *Y*
X_Y	*X* is *Y’s* public key
[\$W,X,m]	key *X* is insured by *W* for up to *m* amount (insurance certificate)
$[\$W, X, m]_{X''}$	Insurance certificate signed with key *X’* (binding certificate)
$X_W\$_m W$	*W* signed public key with $X_W m$ amount (insurance certificate)
(A) → *(B)*	“*says*” relation between entities A and B
HE	Hybrid Element
NWE	Network Element
CSR	Recipient Secure Element
SSE	Sender Secure Element
PvK	Private Key
PbK	Public Key

In literature work, the insurance logic is an extension of the delegation logic of Lampson et al. [159] to strengthen authentication in a large-scale system such as distributed system. Therefore, this insurance logic method suits the IoT environment presented in this paper.

Insurance keys are assumed to be verified and easy to accomplish [143]. The user of the system is called an insurer and has a very important role. Keys are issued by users called issuers. These users assume that those keys are insured and known by themselves. In this paper, we do not require that all users must know all insurers’ key. Rather, all insurers are credited with the ability to determine other insurer’s keys. Therefore, insurers’ keys should be very protected and remain accessible within insurers only. Notably, insurers that are deemed unscrupulous insurers, severely and too often misbehave. If detected by the Security Framework Bridge (SFB), those unscrupulous insurers must be deleted, banned, or punished.

Trust in insurance adds a trustworthy label to the entities involved in the transaction by assuming that the latter entity is trustworthy than a key signed by those same entities.

Nonetheless, there is no absolute guarantee that those trustworthy entities could still be unscrupulous certificates' insurers. Table 5.1 illustrates the insurance logic annotation used in this paper.

5.2.2 Related Work

Designing Resilient IoT Devices

Most work in the literature refers to modelling and designing resilient systems to be able to ensure computational continuity in cross-IoT environment. N. Hossein Motlagh et al. attempted to solve complex tasks such as managing applications and services running in IoT space since the operations of IoT devices and services are tightly reliant on the IoT space environment characteristics, number of IoT devices as well as the type of deployed technologies [160]. The approach proposed a concept with specifications that could foster digital integration of heterogenous cyber physical systems in cross IoT environment.

Saddik et al. [161] proposed a derived strategy from digital twin paradigm to mirror the image of a physical process modelled to match the process taking place in a physical device. The digital twin strategy effectively manages the complexities of cyber-physical systems. This work did not extend the strategy to data transmission between diverse components of the cyber physical system.

Tao et al. [162] proposed digital twin models enabling seamless data communication between the so-called physical reality. The physical reality encompasses data measured from in built smart sensors to perform monitoring, understanding and optimization of the cyber physical system. The proposed models did not specify the practical benefits of analytics raw information accessed through sensors data information.

The upgraded design of the Security Engine needs to ensure resilience to cyber-attacks while gathering data, selecting the secure data path and securely transfer its processing profile.

RF Fingerprinting

Various proposed works from the literature present many different IoT security techniques. Most common techniques focus on hardware identification, such as the Network Interface Card (NIC) transmitting an IEEE 802.11 frame.

He et al. [163] discusses techniques related to RF fingerprinting for addressing challenges with localization-based approaches namely localization accuracy, network time delays, radio resources availability, signal level.

Based on network traffic analysis, the reference [166] presented a fingerprinting technique for wireless devices by observing their emitting traffic on local area network (LAN). The method requires a dense traffic to capture network behavior to formulate signature for each device. But IoT network traffic is very minimum to generalize this technique to an entire IoT environment. In Addition, IoT networks have already a brownfield of legacy devices deployed and still active.

Numan et al. [164] proposed a network interface card fine time measurement technique applied to machine learning method for mobile device indoor localization. The technique is limited to use of precise time measurements data as a main feature characteristic input to the machine learning model.

He et al. [165] proposed a technique based on time of arrival recorded on a cellular network for improving the accuracy localization of the user on the roam. The technique did not extend to heterogeneous networks like IoT networks.

Wu et al. [166] proposed a technique called PARADIS that collects presents hardware imperfections data information, then performs a machine learning based fingerprinting to identify the distinctive NIC. Nevertheless, this technique relies heavily on the performance of the chosen machine learning classification tool.

Based on protocols, Baldini et al. [167] proposed a method to fingerprint device based on common similar use transmission protocols given different devices transmitter. Their approach is based on the behavior of the devices for the observed same protocol, but it cannot be applied to IoT network since the IoT environment itself benefits from heterogeneous protocols, which will take longer and will consume a lot of resources to process all the data information for fingerprinting purposes only.

Based on network traffic analysis, Miettinen et al. [168] presented a fingerprinting technique for wireless devices by observing their emitting traffic on local area network (LAN). The method requires a dense traffic to capture network behavior to formulate signature for each device. But IoT network traffic is very minimal to generalize this technique to an entire IoT environment. In Addition, IoT networks have already a brownfield of legacy devices deployed and still active.

Therefore, a solution combining IoT based PKI and IoT fingerprinting will enhance security for IoT networks [169]. Radhakrishna et al. [170] proposed a mechanism based on location channel randomness pairing. The work has been tested only on their prototype and required implementation on all devices.

Public Key Infrastructure (PKI)

The PKI is the manager of the required key for both public users and private users. However, any user who intends to prove ownership of a key must hold a certificate verifiable by a Certificate Authority (CA).

The most common implementation of PKI is based on the X.509 standard [171] that verifies an entity's ownership of a CA's issued public key on the request. In this process, the verifying entity keeps the root certificate and trusts the CA if the certificate is successfully verified. Cooper et al. [172] have introduced PKI as a front-line security mechanism in the context of cryptography, where communication and data security of the internet are threatened. X.509 based PKI standard research problems are as follows:

1- Lack of redundancy: Single point of failure for CA-based PKI.

2- Lack of traceability: CA-based PKI does not offer transparency.

3- Lack of recoverability: CA-based PKI must revoke certificates only option when found CA compromised by rogue attack therefore the upgrade design would offer recoverability.

The attempt to propose a solution by Laurie et al. [173] contributes to Google's Certificate Transparency (GCT) project. This approach offers monitoring and auditing capabilities to each CA domain server for the newly added certificates [174]. Moreover, this approach adds transparency to the current PKI architecture but does not guarantee the existence of illegitimate forged certificates in the certificate vault logs [175].

In the context of IoT networks, authors in [176] investigate session private/public key distribution between smart home management systems and IoT devices. In this approach, CAs pair only the sent or received key from the light source device, including IoT devices that are out-of-band of communication.

Millen et al. [177] suggested that insurance can be used in distributed and large-scale systems to mitigate individual risks inherent throughout the authentication procedures. The advantage of their PKI's approach lies in bringing trust relationships and insurance together to provide confidence in the secured authentication processes. Although their work offers significant examples of how it could work, an implementation of it has not been provided. Therefore, in this paper, we will implement and enhance the recommended PKI approach.

It is worth noting that an insurance certificate issued by an insurer's entity is ultimately considered as a kind of authorization certificate. Therefore, an insurance certificate could eventually be implemented or serve as an upgrade to an existing Simple Public Key Infrastructure (SPKI) certificate system [178].

5.3 Proposed System

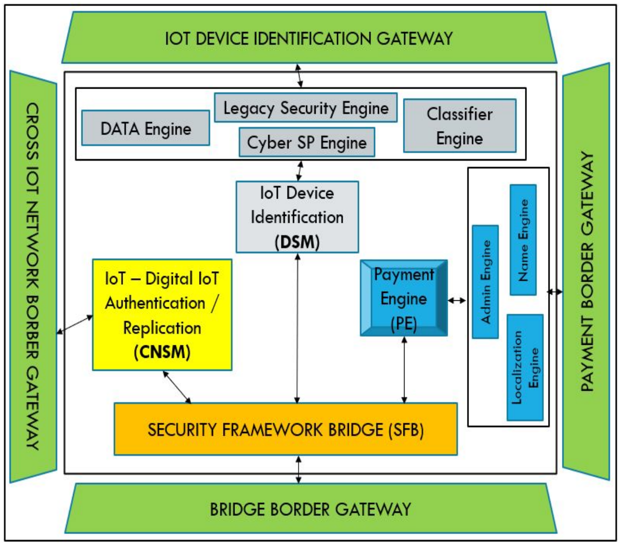

IOT DEVICE IDENTIFICATION GATEWAY
CROSS IOT NETWORK BORBER GATEWAY
PAYMENT BORDER GATEWAY
DATA Engine
Legacy Security Engine
Cyber SP Engine
Classifier Engine
IoT Device Identification (DSM)
IoT – Digital IoT Authentication / Replication (CNSM)
Payment Engine (PE)
Admin Engine
Name Engine
Localization Engine
SECURITY FRAMEWORK BRIDGE (SFB)
BRIDGE BORDER GATEWAY

Figure 0.1 Proposed Cyber Physical Security Engine CPSE.

The proposed upgraded security engine is featuring a cross-domain device replication that can replicate and securely transfer the profile of the Internet of Things (IoT) user device to another. The Enhanced RF fingerprinting by device location module named DSM – Device Secured Module - offers protection through device identification for IoT hardware and radio communications, while the CNSM – Cross Network Secured Module - offers continuation of the security engine service when the IoT user device is facing interruption of services through service authentication for protocols and applications exchanges with integrated location-based services. Both CNSM and DSM are designed to be embedded into a Security Engine (SE).

5.3.1 System Overview

The proposed system architecture, as shown in Figure 5.1, is composed of two main blocks: Block 1 is Device-Secured Module (DSM) and block 2 is Cross Network-Secured Module (CNSM).

According to the system architecture upgraded design, an identification request to DSM by an IoT device before any user interface transaction is to be permitted. Then, this IoT device is prepared to forward the control to the user interface application assuming all communications channels are established.

The IoT device receives the previous Public Identification Key (PbIDK) from the user through User Interface (UI) application, and the IoT device checks its status of identification permission to allow the user to gain access to UI's application requested services.

DSM Module

The main objective of the DSM module is to build up an effective and secure IoT device environment through device identification mechanisms such as RF Fingerprinting [179]. Moreover, DSM offers the first stage of protection through RF fingerprinting identification depending on the IoT environment and the service requested by the application of the user interface (UI). DSM is basically in charge of data acquisition, feature extraction and classification.

Data acquisition is performed by an acquisition submodule, which acquires and digitizes radio signals from connected IoT wireless devices [180]. DSM performs data acquisition in either active or passive mode [181].

RF features extraction [182] is the next process for RF Fingerprinting, which generates characteristic attributes from the raw signal emitted by the IoT devices. This activity of features extraction is governed by any hypothetical extraction concept that minimizes the input dimension to achieve the efficiency of the extraction process. Therefore, the length of the feature vector will be reduced without missing elements needed to perform the next step with the classification process.

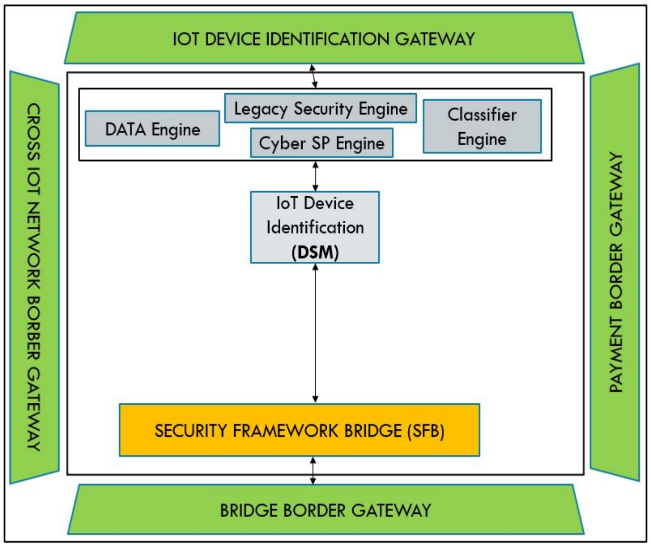

IOT DEVICE IDENTIFICATION GATEWAY
CROSS IOT NETWORK BORBER GATEWAY
PAYMENT BORDER GATEWAY
DATA Engine
Legacy Security Engine
Cyber SP Engine
Classifier Engine
IoT Device Identification (DSM)
SECURITY FRAMEWORK BRIDGE (SFB)
BRIDGE BORDER GATEWAY

Figure 0.2 Device security Module (DSM) system module of CPSE.

The best description of classification [183] is the process initiated to perform a task on a trained network subjected to respond when an input vector like a learned vector is presented. The literature indicates that most classifiers follow an approach initially called Bayesian [184] classification to provide a common solution to pattern classification problems. In our proposed SE, the DSM achieves level 1 protection of the IoT networks and environments, as shown in Figure 5.2.

CNSM Module

The CNSM is deployed by DSM to validate the claim of device identification before proceeding to payment realization for instance. The IoT device gets its Public Key Identification PbKID from User Interface (UI). UI checks the validity of his permission request for the certificate. The validation process’s next step involves the Security Framework Bridge (SFB).

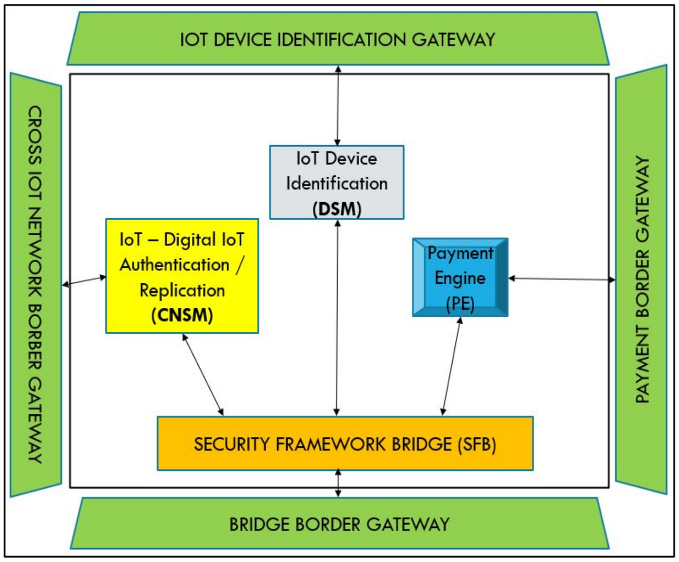

IOT DEVICE IDENTIFICATION GATEWAY
CROSS IOT NETWORK BORBER GATEWAY
PAYMENT BORDER GATEWAY
IoT Device Identification (DSM)
IoT – Digital IoT Authentication / Replication (CNSM)
Payment Engine (PE)
SECURITY FRAMEWORK BRIDGE (SFB)
BRIDGE BORDER GATEWAY

Figure 0.3 Cross Network Security Module (CNSM) system module of CPSE.

The CNSM connects to the Payment Engine (PE) through SFB for payment proceedings and realization.

The CNSM responds to DSM through SFB to complete validating claims of IoT devices successfully compliant to protection level 1.

In our upgraded design, CNSM acts as a replication authority for authentication of digital (replicated) IoT devices.

CNSM acts as a protection level 2 in our proposed SE system, as shown in Figure 5.3.

Security Framework Bridge (SFB)

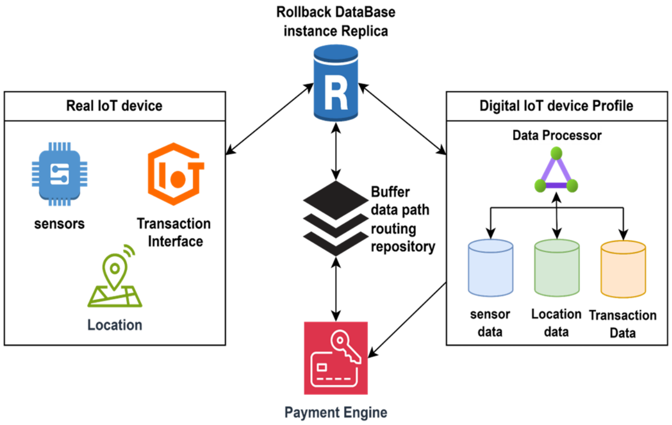


Figure 0.4 Digital Replication feature of CPSE.

The Security Framework Bridge (SFB) functions as Trust Third Party (TTP). IoT devices request a digital certificate through forwarded request by CNSM. SFB includes two on demand functions that are essential for the integrity of the SE system. These functions are integrated into the SFB modules as rollback and buffer.

Rollback is essential to the PE transaction recovery if there is any issue with the payment transaction completion. In the upgraded SE design, Rollback is featured with IoT device profile digital replication as shown in Figure 5.4.

Buffer is offering the property of caching transactions for the SFB as a second volatile temporary read only memory. It can be used to speed up the processing time for the payment transactions and provide the additional resources that might be needed in case a transaction log exceeds the original length and size.

CNSM directly communicates with Buffer in case SFB fails to continue to operate.

Similarly, PE directly interfaces with Rollback.

SFB confirms DSM protection level 1 status (i.e., success or failure). Then, after following its upgraded algorithm, SFB confirms CNSM deployment and protection phase 2 status (i.e., success or failure).

In case of any potential failure at phase 2, SFB pulls the routing data path records stored in Buffer and validates with Rollback the destination IoT device for profile transfer for continuation of payment transactions.

The next task for SFB is to send session key to UI for decryption while monitoring Rollback and Buffer.

In our proposed CPSE system, SFB acts as a bridge to DSM, CNSM, PE and UI, as shown in Figure 5.5.

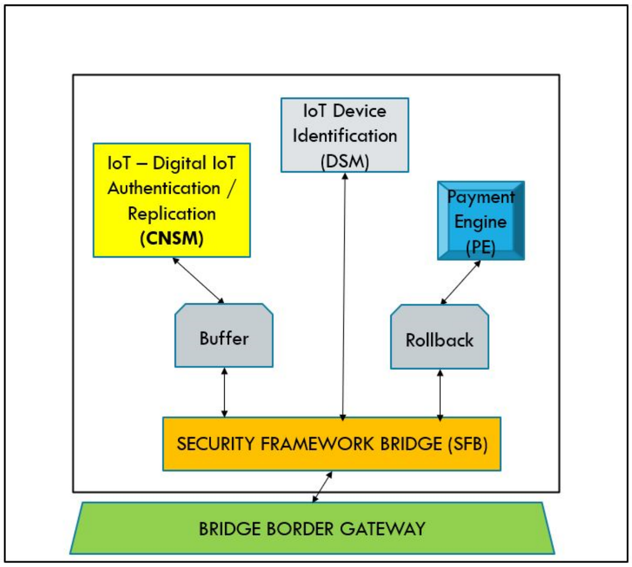

IoT Device Identification (DSM)
IoT – Digital IoT Authentication / Replication (CNSM)
Payment Engine (PE)
Buffer
Rollback
SECURITY FRAMEWORK BRIDGE (SFB)
BRIDGE BORDER GATEWAY

Figure 0.5 Security Framework Bridge (SFB) system module of CPSE.

5.3.2 System Workflows

Certificates and Keys Exchange Scheme

The challenge in this large-scale system is the certificate exchange to distribute to many IoT devices following the PKI infrastructure. The exchange scheme is shown in Figure 5.6 followed by the exchange mechanisms. In the key exchange system, the IoT device makes a request to the CNSM module through the IoT network border gateway. This latter can be connected to any third-party PKI distribution infrastructure such as a public key distribution system in blockchain [185]. It is worth noting that the CNSM module in Figure 5.3 represents the abstract of the SE system.

The public key for IoT devices stores additional information about the device ID, OS version and Processing Unit (PU) capabilities. Moreover, one-time communication session key's property can reduce the risk of key's leakage such as reverse engineering key retrieval, since the session key is completely updated from IoT device buffer memory after the set timeout has expired.

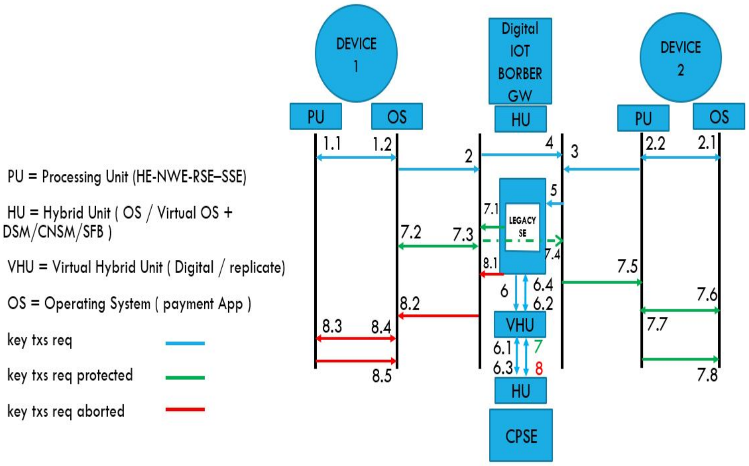


Figure 0.6 Proposed Certificates and Keys exchange scheme.

(1) Start

(1.1) Device 1 requests UI credentials (1.2)

(2.1) Device 2 requests UI credentials (2.2)

(2) Device 1 Application Interface adds UI credentials to HU/DSM

(3) Device 2 Application Interface adds UI credentials to HU/DSM

(4) HU/DSM requests public key of device 2

(5) Protection Engine receives public key of device 2

(6) HU/CNSM validates public key

(6.1) HU/CNSM confirms DIGI_IOT (IoT device replicate)

(6.2) HU/CNSM shares IOT_DST_RT (IoT device replicate destination route) with SFB

(6.3) HU/SFB confirms function IOT_DST_RT in SFB/ROLLBACK

(6.4) HU/SFB reloads / updates function DATA_PATH in SFB/BUFFER

(7) HU/SFB authorizes / aborts (8) sessions between device 1 and device 2.

(7.1) Confirmation session authorization to device 1

(7.2) Device 1 generates session key

(7.3) HU/SFB encrypts the session using device 2 public key

(7.4) Confirmation session authorization to device 2

(7.5) PU/Device 2 pulls device 1 public key information

(7.6) Device 2 application interface receives session key to decrypts

(7.7) Device 2 decrypts the session key with its private key

(7.8) Service request transferred to border gateway

(8.1) Confirmation session abortion to device 1

(8.2) Device 1 generates rollback session key

(8.3) PU/Device 1 revokes device 2 public key information

(8.4) Device 1 application interface release session

(8.5) OS/Application notifier update

(9) Stop.

Insurance and Trust Scheme

Insurance and trust scheme refers to authentication in large-scale distributed systems according to LABW logic [159], interprets a certificate as a statement representation. There are two paths to building insurance: the shorter path and the longer path.

- Shorter path

The shorter path in insurance logic works without need to build trust with the other users of the system. It is described as follows:

(1) AND $[D_1, K_{D_1}]_{K_{D_2}}[\$D_3, K_{D_2}]_{K_{D_3}}$

(2) $K_{D_3} \Rightarrow D_3$

(1) is interpreted as

(3) K_{D_2} says $K_{D_1}\Rightarrow$ AND says \$$D_1 K_{D_3} K_{D_2} D_3$

(1) AND (3) give

(4) says $D_3 K_{D_2}$\$$D_3$

(5) K_{D_2}\$$D_3$

(2) AND (5) give

(6) $(K_{D_1}\Rightarrow$)\$$D_1 D_3$

With the use of the shorter path here, we can conclude that link relation between and .D_3 has insured the $D_1 K_{D_1}$

Moreover because (6) may not apply since:

(7) $K_{D_1}\Rightarrow$ is true.D_1

- Longer path

The longer path in insurance logic combines trust and insurance in the same path. It is described as follows:

(1) $[D_4, K_{D_4}]_{K_{D_1}}$

(2) if D_1 says $(K_{D_4}\Rightarrow$) then D_4 $K_{D_4}\Rightarrow D_4$

(3) K_{D_1} says $(K_{D_4}\Rightarrow D_4)$

(4) $(K_{D_1} \Rightarrow D_1) D_3$, OR D_3\$

(4) is true

(5) $K_{D_4} \Rightarrow D_4$

If is damaged by any user device, thenK_{D_4}

(5) is false

Trust returns to since it was the first trustworthy introducer with no ever damaged or reported compromised.$D_1 K_{D_1}$

Cyber Physical Resilient Security Engine's Proposed Algorithm

Algorithm 1 presents the cyber physical resilient design of the Security Engine (SE) representing the virtual connection of an IoT secure environment connecting to four gateways (i.e., the IoT device gateway, the IoT network border gateway, the payment boarder gateway, and the bridge border gateway). The algorithm I for the CPSE has a beginning phase with the module in charge of the device security level described as level 1 and named DSM. DSM protection phase 1 starts when the IoT device starts transmitting radio signals.

The device's RF fingerprinting details need to pass the stage of full transmission for the DSM to make a request to the next level module using the positioning details. DSM requests confirmation of GPS service resource availability and demands SFB to deploy CNSM.

ALGORITHM 1 CYBER PHYSICAL RESILLIENT SECURITY ENGINE CORE ALGORITHM

```
Input: S(V).
V={M∪K},
M={DSM,CNSM,SFB,PE},
K(n,m)={PvIDK1,PbIDK2,PvIDKn....PbIDKm}.
Output: IoT destination protection phase state S.

1 : DSM protection phase 1 STARTS
1.1 : If device's RF FINGERPRINTING passed then DSM requests confirmation of Native GPS service resource availability,
1.2 : DSM updates latest device's location with geolocalisation code precision t (seconds taken to update gps information)
1.2.1 : DSM requests SFB to deploy CNSM
1.3 : else DSM requests SFB protection phase 1 at p ( time taken to fail phase 1) to be aborted.
2: CNSM protection phase 2 STARTS
2.1: if CNSM validates Device's ID claims by ISSUING SIGNED ID CERTIFICATE using USER INTERFACE PbIDK then CNSM replies to DSM copy of USER INTERFACE PbIDK for user payment application processing through PE
2.1.1: CNSM sets DATA_PATH using SFB / BUFFER
2.1.2: CNSM saves IOT_DST_RT using SFB / ROLLBACK
2.2: SFB bridges DIGI_IOT to USER INTERFACE
2.2.1: SFB sends session KEY to USER INTERFACE with SIGNED CERTIFICATE through PE for user to decrypt with PvIDK
2.3 : else CNSM requests SFB protection phase 2 at p' (time taken to fail phase 2) to be aborted.
3: SFB confirms Protection Phase 1 AND Protection Phase 2 SUCCESS then forwards ACK to PE to proceed with user payment transaction
3.1: if user failed decryption with PvIDK then PE request SFB to abort payment or service request
4: else if SFB returns CNSM Protection Phase 1 status and DSM Protection Phase 2 status then update PE to proceed with offline user payment or service request.
```

The SFB routes the requests made from DSM to CNSM and vice versa. Essentially, CNSM oversees the protection phase 2. Protection phase 2 starts with process validation and PKI keys generation and exchange processes.

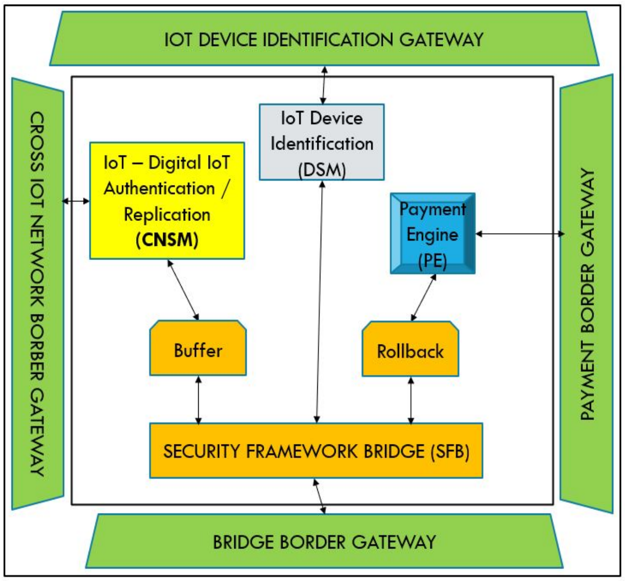

IOT DEVICE IDENTIFICATION GATEWAY
CROSS IOT NETWORK BORBER GATEWAY
PAYMENT BORDER GATEWAY
IoT Device Identification (DSM)
IoT – Digital IoT Authentication / Replication (CNSM)
Payment Engine (PE)
Buffer
Rollback
SECURITY FRAMEWORK BRIDGE (SFB)
BRIDGE BORDER GATEWAY

Figure 0.7 Security Framework Bridge (SFB) system with payment protection.

CNSM validates IoT device's identification claims by issuing signed certificate using user interface PbIDK, then CNSM replies to DSM with a copy of user interface PbIDK for user payment application processing through Payment Engine (PE).

The overall session is monitored by the SFB. This latter sends session key to user interface with signed certificate through PE for user to decrypt with PvIDK.

SFB confirms whether protection phase 1 and protection phase 2 succeeded, then forwards ACK message to PE to proceed with user payment transaction, as shown in Figure 5.7 with the core payment modules components.

Device Replication Modeling Block Diagram

Signal generation can be done using GNU radio SDR software for recording IoT device transmitted raw signal at any different dedicated use mode of operation, such as during phone calls, mobile date payment, and sending MMS and SMS, as shown in Figure 5.8. Each IoT device can operate between a range (890 MHz to 3500 MHz) of LTE / 5G frequency of the carrier cellular mobile network

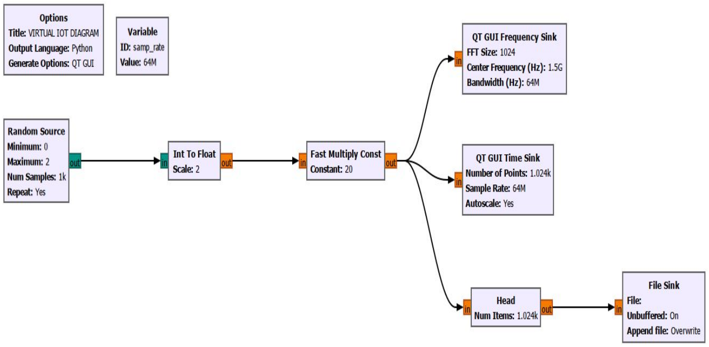


Figure 0.8 Proposed CPSE GNU representation block diagram.

following orthogonal frequency-division multiplexing (OFDM) propagation. The RF specifications make the design of our proposed Security Engine and the replicated IoT device compatible with the arduino test experimental board GSM MKR [186], as illustrated in Table 5.2 and Table 5.3.

Table 0.2 MKR 1400 Specification

Device Model	**32-bit development board**				
connectivity	2G / 3G				
Chipset	ATSAMD21				
Clock	48 Mhz				
Memory	256 KB FLASH		32 KB SRAM		
interfaces	USB	SPI	12C	12S	UART
VOLTAGES	5V INPUT		3.3V OPERATING		
Pinout	22 INPUT		12 PWN	7/1 ANALOG	
Dimensions	67.64 x 25 mm				

Table 0.3 Replicated Specification

Replicated Model	**64-bit development board**
connectivity	HOST CONNECTIVITY (5G / Gigabit 1Gbps)
Chipset	HOST CHIPSET/BIOS
Clock	HOST VIRTUAL MACHINE (3.3 Ghz)
Memory	HOST VIRTUAL MEMORY (2Go)
interfaces	HOST VIRTUAL INTERFACES (USB3.0)
VOLTAGES	HOST VIRTUAL VOLTAGE OPERATING ENVIRONMENT
Pinout	Host Virtual Digital Out
Dimensions	Host virtual machine

5.4 Results and Discussion

5.4.1 Experimental Setup

For this experiment, Table 5.4 presents our experimental setup. Our goals are to obtain significant results, and we assume signal degradation reduction due to noise is performed at the data acquisition stage using high-quality signal measurement [187]. For the benefits of the simulation, we simulated the replicated IoT device using VirtualBox hypervisor and we used Software Defined Radio (SDR) software GNU Radio [188] as a signal processing engine in the virtual machine environment while the device hardware provides the RF front end. GNU Radio provides an extensive library of processing blocks running functions and algorithms, such as encoding, decoding, mixing, filtering, equalizing and packet handling. GNU Radio modeling is tested by an IoT GSM Arduino board for running the

CPSE algorithm sub-routine in passive mode, since we interact with the system by SMS messaging. Successively users will send messages to the CPSE system through the MKR board processing user identification.

The passive mode of data acquisition is used in our experimental setup since Global System for Mobile Communication (GSM) standard group of previous works had applied passive signal of data acquisition for device identification purposes for cellular telephones communicating with real or emulated base node stations [186].

GSM tests are conducted using the arduino board MKR 1400. The MKR board has an inbuilt GSM modem, and we attached a

Table 0.4 Experimental Setup

item	description
Virtual host / MKR 1400	32-bit development testing board
Virtual host Connectivity / Antenna	Dipole Pentaband Waterproof Antenna GSM 850 / 900 / 1800 / 1900 MHz and UMTS bands
SIM Card	Arduino SIM
Virtual host running IDE	ARDUINO IDE 1.8.19
VirtualBox running PC	Windows 10 pro OS, 8 GB, 512 GB, core i5 vPro

Table 0.5 IoT Device Interaction

Test #	Round	Status	DSM Phase 1		CNSM Phase 2		SFB		SE	CPSE	Timestamp	
			IoT	Digital IoT	IoT	Digital IoT	IoT	Digital IoT	IoT	Digital IoT	SE	CPSE
1	4	success (%)	4 (100)	4 (100)	4 (100)	4 (100)	4 (100)	4 (100)	8 (100)	12 (100)	21:04:10.149	01:23:10.050
		fail	0	0	0	0	0	0	0	0	-	-
2	7	success (%)	5 (71)	6 (86)	5 (71)	6 (86)	5 (71)	6 (86)	10 (48)	18 (88)	21:04:10.634	01:23:10.535
		fail	2	1	2	1	2	1	4	3	21:04:22.588	01:23:22.489
3	13	success (%)	10 (76)	11 (84)	9 (69)	10 (76)	9 (69)	10 (76)	19 (49)	31 (79)	21:05:44.826	01:24:44.727
		fail	3	2	4	3	4	3	7	8	21:09:16.298	01:28:16.199

dedicated GPS module for computing the GPS coordinated positioning metrics as longitude, latitude, and accuracy [187].

Table 0.6 Clients and Key Exchange

Test #	Round	Status	DSM Phase 1		CNSM Phase 2		SFB		SE	CPSE	Timestamp	
			IoT	Digital IoT	IoT	Digital IoT	IoT	Digital IoT	IoT	Digital IoT	SE	CPSE
1	4	Success (%)	3 (75)	4 (100)	3 (75)	4 (100)	3 (75)	4 (100)	9 (75)	12 (100)	20:22:04.709 -> AT	00:41:04.610 -> AT
		fail	1	0	1	0	1	0	3	0	20:22:15.193 -> ERROR	-
2	7	success (%)	5 (71)	7 (100)	5 (71)	7 (100)	5 (71)	7 (100)	15 (75)	21 (100)	20:22:16.189 -> OK	00:21:16.090 -> OK
		fail	2	0	2	0	2	0	6	0	20:22:16.189 -> ERROR	-
3	13	success (%)	9 (69)	13 (100)	10 (76)	13 (100)	9 (69)	13 (100)	28 (71)	36 (92)	20:22:16.113 -> OK	00:21:16.414 -> OK
		fail	4	0	3	0	4	0	11	0	20:22:16.181 -> ERROR	-

The MKR GSM 1400 is a great option for GSM connectivity development. The MKR board is using the popular Arm Cortex-M0 32-bit SAMD21 processor. The development board also features the powerful u-blox SARA-U201 module and the ECC508 crypto-chip for security.

We assume that in our setup individual users through the user interfaces of the virtual host, make their own unsupported decisions as to the thrusted-ness of the certificates' introducers. Instead, we use the internal MKR SSL management to comply with the MKR board resource constraint.

In this scenario, we do record data throughout the experiments on the virtual environment hosting the replicate data profile of the replicated IoT device. This testbed represents the replicated IoT security device turned into a cyber physical resilient security engine.

In our current experimental setup, we made the abstraction of possible generation of audited key [188] – [189] to privilege faster computation and keep this latter within the circumscription of the proposed cyber physical resilient Security Engine (CPSE).

Table 0.7 CPSE Performance Metrics

Test #	**Round**	**Simulation**		**DSM** **Phase 1**		**CNSM** **Phase 2**		**SFB**		**SE**	**CPSE**
				IoT	**Digital IoT**	**IoT**	**Digital IoT**	**IoT**	**Digital IoT**	**IoT**	**Digital IoT**
1	4	timestamp	start	21:03:27.445	01:23:03.346	21:04:10.634	01:23:10.535	21:04:46.645	01:26:02.546	21:04:46.645	01:26:02.546
			stop	21:04:10.149	01:23:10.050	21:04:46.117	01:23:46.018	21:04:46.646	01:26:02.547	21:04:47.641	01:26:03.542
		Cross Environment Sensors		31		32		32		32	
2	7	timestamp	start	21:05:30.159	01:25:54.258	21:05:44.770	01:23:44.869	21:05:57.877	01:27:13.778	21:05:57.879	01:27:13.780
			stop	21:05:44.826	01:26:44.925	21:05:57.328	01:25:60.352	21:05:57.878	01:27:13.779	21:05:58.903	01:27:14.804
		Cross Environment Sensors		32		32		33		33	
3	13	timestamp	start	21:07:11.147	01:27:35.246	21:07:31.214	01:28:34.435	21:08:11.075	01:30:06.025	21:08:12.652	01:30:30.356
			stop	21:07:24.529	01:28:24.628	21:07:36.894	01:28:39.918	21:08:11.619	01:30:07.519	21:09:29.323	01:31:13.685
		Cross Environment Sensors		33		33		33		33	

5.4.2 Results

In the subsequent tables' result, test 1 shows 3 attempts from user toward the CPSE, test 2 shows 5 attempts and test 3 shows 10 attempts respectively. The timestamp shows the recorded time duration of the test when the security modules DSM, CNSM and SFB successfully respond to the user attempts respectively.

Table 5.5 illustrates IoT device interaction with IoT user application and device authentication results.

Table 5.6 represents PKI key exchange results.

Table 0.8 IOT Devices Comparison

Device Model	**IoT 1**	**IoT 2**	**SE MKR 1400**	**Proposed Cyber Physical CPSE**
Platform	ArmV8	32 bit ARM	ATSAMD21 32 bit	Virtual Host (32bit/64bit)
Open-Source model	no	no	yes	yes
On device user SmS	optional	optional	yes	yes
On device SSL	no	no	yes	yes
External encryption resource	required	required	no	no
IoT device recovery	no	no	no	yes
IoT Devices interaction	yes	yes	yes	yes
Mobile Device companion	required	required	no	no
GPS Dual frequency	yes	yes	yes	yes

Table 5.7 represents the performance metrics recorded for the proposed CPSE system.

5.4.3 Discussions of Results

Our proposed cyber physical resilient security engine CPSE's features, tested on virtual host based on the arduino MKR1400 are compared against security SE MKR 1400 [190], IoT 1 and IoT 2 devices in Table 5.8. Unlike other devices, our proposed cyber physical resilient SE does feature IoT device recovery and does not require additional device companion to be able to offer user device interaction and SSL authentication. IoT device recovery capability is enabled thanks to device replication feature as illustrated in Figure 5.4.

In Table 5.7, we recorded the cross environment sensor levels in terms of temperature of the MKR physical board extended to the digital IoT machine ranging from 31 degrees Celsius at test 1 round 4, to 32 degrees at test 2 round 7, to 33 degrees at test 3 round 13. Since at phase 1, DSM is responsible for user authentication after successful device interactions, the average response time of the digital IoT device is around 7 seconds faster

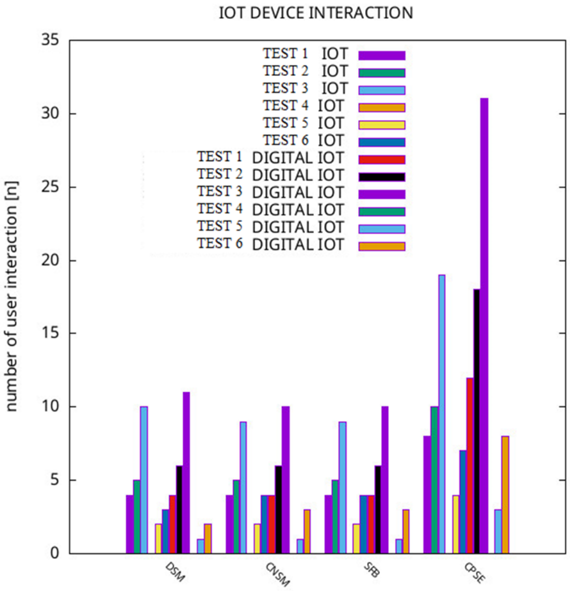


Figure 0.9 Number of interactions with CPSE

than the physical device in reference to the timestamp recorded in Table 5.5 and Table 5.6. Our approach to device identification with recovery though digital replication extends the use of the GSM module built into the MKR board. This latter uses the GSM library to send and receive SMS in an effective way. Moreover, the GSM library allows the proposed system to connect to the internet through the GPRS networks by using web clients for the requester (SMS sender) and server for the responder (SMS receiver). In addition, the uniqueness of the user mobile number helps to bind the user application requests with the api level capability of the requested services, as shown in Figure 5.9. Figure 5.9 shows that the digital IoT device outperforms the physical IoT device at test 3.

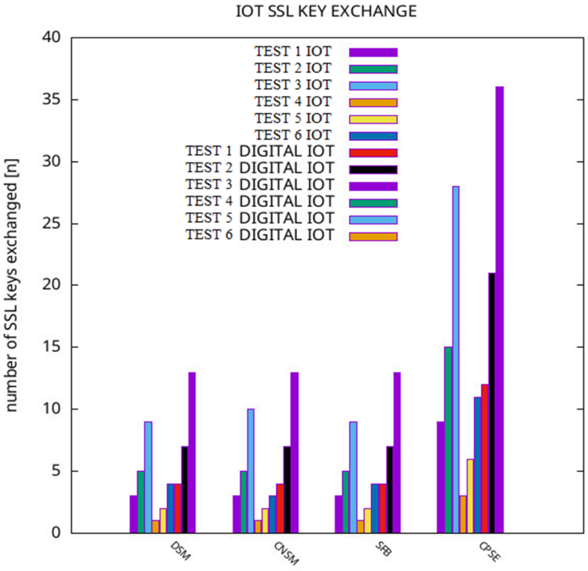


Figure 0.10 Number of SSL key exchange with CPSE

Our approach to authenticate the users using SSL management of the MKR board brings SSL clients in the based class for all GPRS SSL client-based calls. The SSL is not called directly but invoked whenever the system uses a function withing the MQTT broker client that relies on it. MQTT client library implementation is ported to support ESP32/S2/S3/C3, WT32_ETH01 (ESP32 + LAN8720), ESP32 using LwIP ENC28J60, W5500, W6100 or LAN8720. The same library supports TLS/SSL for MQTTS clients, which is an ideal choice for our proposed SE and makes it compatible with most communication protocols running on IoT devices. The proposed system allows creating a client that always connects SSL to the specified IP address using the GPRS mobile operator sockets and port, even if client.connect() is used instead of client.connectSSL(). Our implementation helps us to find that it is useful even if we have a library that accepts only plain client, but we want to force the client to use SSL, keeping the same method names of the non-SSL client. This makes our proposed system secure for any user client IoT connected devices or applications, as shown in Figure 5.10. Figure 5.10 shows that more SSL keys have been exchanged by the digital IoT at test 3 than the physical IoT device.

Our approach based on Public Key Infrastructure in the large-scale environment of devices like IoT, is useful if the certificates are self-signed by users called introducers. There are no needs for a third trust party (TTP) such as certification authorities in the current form of our proposed approach. Users of the proposed system can decide who are trustworthy certificates' introducers or who is not. Our system does not follow any hierarchy path of insuring the certificates' introducer. In such a case, some users will or will not certify other's keys and in return, such users may or may not have their own keys insured by others.

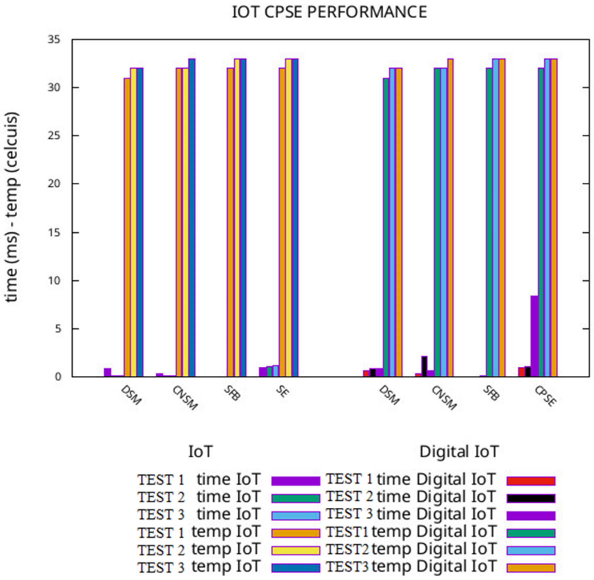


Figure 0.11 CPSE response time performance metrics

Figure 5.11 depicts the overall time, and the performance of the proposed system taken into completing phase 1 and phase 2 till the final cycle. We noticed that all modules of the Digital IoT outperformed the physical IoT modules in test 1, 2 and 3.

5.5 Conclusion

In this paper, we presented a new approach towards cyber physical IoT security relying on a cross-domain device replication that can replicate and securely transfer the profile of the Internet of Things (IoT) user device to another. For any application to IoT, the security of the transaction is maintained by adding a cybersecurity abstraction layer to offer continuation of the security engine service when the IoT user device is facing interruption of services. Our experiments on testing the IoT operating system on the virtual machine acting as a digital IoT device show a communication success rate of 80 percent between all Security Engine components called modules on Physical IoT devices and 90 percent on replicated virtual IoT devices while improving the simulation running time and offering a recovery option. Our research work can be recommended for resiliency option for secure anonymity, privacy, and non-repudiation user application purposes using a registered IoT device where the IoT device benefits from using the proposed CPSE for computational continuation in optimal performance maintaining integrity of the transferred data to the virtual digital IoT device.

As future work, some improvement needs to be addressed in terms of computational resource limits of our proposed CPSE system since the testing development virtual machine perhaps while operating had shown some dependency while relying on the host computer specification. Furthermore, we can repurpose the current design of the CPSE to target large-scale environments, such as IoT cloud, crypto exchanges and blockchain without limitations on artificial intelligence (AI) models since our current system can now feature an energy-efficiency model for AI acceptable computations.

Acknowledgment

The authors wish to thank Dr. Franjieh El Khoury for her constructive comments and the proofreading of this paper.

CHAPTER 6 CONCLUSION

6.1 Summary of the Work

The work of our thesis was enriching because each part of the thesis indicating the writing of an article allowed us to:

- Designing an IoT Architecture for Mobile Payments
- Design a common security framework for IoT devices and payment applications
- To offer a hybrid evaluation of terminals available to users to recommend certain according to application uses and performance.
- Design modular security and test it using Arduino test boards.
- Strengthen modular security and test its continuity of service against physical cyber-attacks.

6.2 Limitations

Mobile payment represents a great opportunity for users, since it appears to be a favorable solution to the payment problem while being mobile. Moreover, it allows users to reduce the costs of sending transactions by considering the user's mobility. However, for some analysts, mobile payment represents a threat to the security of user information and the significant expense of adopting IoT and a mobile payment terminal, especially in developing countries with such technology.

6.2.1 Payment Security in an IoT Environment

Although security is considered in mobile payment, it appears that security has become an important factor in the mobile payment system. Indeed, the risks involved when sensitive user data is used for transactions, involving several entities, can be difficult to achieve. As a result, techniques to prevent the disclosure of payment data should be considered. Thus, this would not make it easy for potential hackers to enter the system.

6.2.2 Cost

Many say that mobile payment is more affordable than other payments. On the one hand, mobile payment is affordable, because it is better to use your own mobile device to make the transaction. On the other hand, it is not affordable because of the purchase cost of acquiring smart mobile terminals, the cost of using mobile data. This is the sad reality of countries with low connectivity, because you will have to pay huge amounts of money to use mobile payment. However, our solution should remedy this situation in the future.

6.2.3 Heterogeneous Management

The transparency of mobile payment gives users a right to inspect the personal information transmitted throughout the sending and receiving process. However, the different actors in this process are not the same. Hence the presence of strong heterogeneity ranging from the mobile data network provider to the payment platform providers through the financial structure affiliated with the user. Thus, passing on this most vulnerable data and information should be done with the certainty of a certain trust in the stakeholders and of knowing on what governance this data depends. These stakeholders are grouped together in a single conglomerate or consortium to assure all users that a single entity oversees all this data, avoiding mentioning any possible contractors.

6.3 Future Improvements

In future work, we will extend this work to the use of **ECC** (elliptic curve cryptographic) and the evaluation of the **power consumption** of each node in a real, unsimulated laboratory environment. In addition, this work can be applied to combat more denial-of-service (DOS) attacks with Artificial Intelligence-based learning optimization techniques and deployment techniques with nominal power consumption.

In addition, if it were possible to allow users to have better access to as many available satellites, this option offers continuous development for the mobile device application industry, especially in mobile payment protection.

The limit of this work will fall into the ability to validate the results against the operation computational resource limit of our test device in Article 2.

In future work, some improvements need to be made in terms of computational time and test board responsiveness, which could perhaps lead to improved results, while also trying to include more **options in the current OS design** to target a **large-scale environment** such as the **Next Generation cloud IoT** *powered by* **Generative Artificial Inteligence (Gen-AI)**, **crypto exchanges,** and **blockchain** in future work.

REFERENCES

[1] Siau, K. and Shen, Z. Mobile communications, and mobile services. Int. J. Mobile Communications, Vol. 1, Nos. 1/2, 2003, 3-14.

[2] Boston Analytics, 2007, A Study of the Mobile Value Added Services (MVAS) Market in India, Retrieved on 24th January 2021 from https://docuri.com/download/mobile-vas-report-2007_59c1cbe1f581710b2861624e_pdf

[3] Katankar V. K. and Dr. Thakare V. M., 2010, "Short Message Service using SMS Gateway", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, pg 1487-1491

[4] Kendall J., Maurer B., Machoka P., and Veniard C., 2012, "An Emerging Platform: From Money Transfer System to Mobile Money Ecosystem", Retrieved on 24th January 2021 from : https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2012/06/ssrnid1830704.pdf

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[6] D. Tapscott and A. Tapscott, "Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World.", Baltimore, MD, USA: Penguin, 2016.

[7] Roman, Rodrigo, Pablo Najera, and Javier Lopez. 2011. 'Securing the Internet of Things (IoT)'." IEEE Computer 44: 51–58. doi:10.1109/MC.2011.291.

[8] L. Hu et al., "Software defined Healthcare Networks," IEEE Wireless Commun., vol. 22, no. 6, Dec. 2015, pp. 67–75.

[9] K. Zheng et al., "Challenges of Massive Access in Highly Dense LTE-Advanced Networks with Machine-to-Machine Communications," IEEE Wireless Commun., vol.21, no. 3, 2014, pp. 12–18.

[10] Jianqi Liu, Jiafu Wan, Dongyao Jia, Bi Zeng, Di Li, Ching-Hsien Hsu, and Haibo Chen, “High-Efficiency Urban Traffic Management in Context-Aware Computing and 5G Communication”, IEEE Communications Magazine • January 2017.

[11] T. Dahlberg, J. Guo, and J. Ondrus, ``A critical review of mobile payment research," Electron. Commerce Res. Appl., vol. 14, no. 5, pp. 265_284, Sep./Oct. 2015.

[12] J. Liu, R. J. Kauffman, and D. Ma, “Competition, cooperation, and regulation: Understanding the evolution of the mobile payments’ technology ecosystem," Electron. Commerce Res. Appl., vol. 14, no. 5, pp. 372_391, Sep./Oct. 2015.

[13] Jeonil Kang, DaeHun Nyang,” A Privacy-Preserving Mobile Payment System for Mass Transit”, IEEE Transactions On Intelligent Transportation Systems, vol. 18, no. 8, august 2017.

[14] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, “When mobile blockchain meets edge computing,” IEEE Commun. Mag., vol. 56, no. 8, pp. 33–39, Aug. 2018.

[15] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, “Intelligent edge computing for IoT-based energy management in smart cities,” IEEE Netw. Mag., vol. 33, no. 2, pp. 111–117, Mar./Apr. 2019.

[16] V. Sivaraman, H. H. Gharakheili, C. Fernandes, N. Clark, and T. Karliychuk, “Smart IoT Devices in the Home: Security and Privacy Implications,” IEEE Technol. Soc. Mag., vol. 37, no. 2, pp. 71–79, Jun. 2018.

[17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, pp. 2347-2376, 2015.

[18] G. N. M. U. K. Theebak and K. C. S. Kumar, "Challenges and Opportunity in Internet of Things (IoT)," 2017.

[19] D. Shrier, G. Canale, and A. Pentland, “Mobile money & payments: Technology trends," in MIT Connection Science's Series on Financial Technology. Cambridge, MA, USA: Massachusetts Institute Of Technology, 2016.

[20] V. Mishra and S. S. Bisht, “Mobile banking in a developing economy: A customer-centric model for policy formulation," Telecommun. Policy, vol. 37, nos. 6_7, pp. 503_514, Jul./Aug. 2013.

[21] Vanesa Daza, Roberto Di Pietro, Flavio Lombardi, Matteo Signorini, “FRoDO: Fraud Resilient Device for Off-Line Micro-Payments”, IEEE transactions on dependable and secure computing, vol. 13, no. 2, march/april 2016.

[22] Xiaoling Dai; Ayoade, O.; Grundy, John, "Off-Line Micro-Payment Protocol for Multiple Vendors in Mobile Commerce," Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference on, vol., no., pp.197,202, Dec. 2006.

[23] DR. S.Manikandan, and J.Mary Jayakodi. (2017). “An emprical study on consumers adoption of mobile wallet with special reference to chennai city.” International Journal of Research - Granthaalayah, 5 (5), 107-115. https://doi.org/10.5281/zenodo.583902.

[24] Sahil Vashisht, Sushma Jain, Ravinder Singh Mann, Guru Arjan Dev, “Software Defined UAV-based Location Aware Deployment Scheme for Optimal Wireless Coverage”, 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress.

[25] “Cellular IoT connections by segment, technology (billion) and IoT connections (billion)“, Https://Www.Ericsson.Com/49da93/Assets/Local/Mobility-report/Documents/2020/June2020-ericsson-mobility-report.Pdf , Visited On 4th November 2020.

[26] J. T. Isaac, Z. Sherali, “Secure mobile payment systems”, vol. 16, no. 3, pp. 3643, May/Jun. 2014.

[27] J. Kaye, J. Vertesi, J. Ferreira, B. Brown, M. Perry, “CHIMoney: Financial interactions, digital cash, capital exchange and mobile money ‘’, in Proc. Extended Abstr. Hum. Factors Comput. Syst., Apr. 2014, pp. 111114.

[28] D. Reuver, E. Verschuur, F. Nikayin, N. Cerpa, H. Bouwman, “Collective action for mobile payment platforms: A case study on collaboration issues between banks and telecom operators”, Electron. Commerce Res. Appl., vol. 14, no. 5, pp. 331344, Sep./Oct. 2015.

[29] Josef Steinbaeck, Christian Steger, Gerald Holweg, and Norbert Druml, " Next Generation Radar Sensors in Automotive Sensor Fusion Systems ", Infineon Technologies Austria AG, Graz, Austria, Graz University of Technology, Graz, Austria, Graz, Austria, IEEE 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF).

[30] J. Lewandowska. (2013). [Online]. Available: http://www.frost.com/prod/servlet/press-release.pag?docid=274238535

[31] Yining Hu, Ahsan Manzoor, Parinya Ekparinya, Madhusanka Liyanage, Kanchana Thilakarathna, Guillaume Jourjonand Aruna Seneviratne, “A Delay-Tolerant Payment Scheme Based on the Ethereum Blockchain” IEEE Access (Volume: 7), 06 March 2019, 33159 – 33172

[32] Babatunde Ojetunde, Naoki Shibata, Juntao Gao, “Secure Payment System Utilizing MANET for Disaster Areas”, IEEE transactions on systems, man, and cybernetics: systems, vol. 49, no. 12, december 2019.

[33] Gartner. Gartner Says 6.4 Billion Connected ‘Things’ Will Be in Use in 2016, Up 30 Percent From 2015. Accessed: May 30, 2022. [Online]. Available: http://www.gartner.com/newsroom/id/3165317

[34] Forbes. 152 000 Smart Devices Every Minute in 2025: IDC Outlines the Future of Smart Things. Accessed: May 30, 2022. [Online]. Available:http://www.forbes.com/sites/michaelkanellos/2016/03/03/152000-smartdevices-every-minute-in-2025-idc-outlines-the-future-of-smart-things/# 34bf983369a7.

[35] N. Zhang et al., “Understanding IoT security through the data crystal ball: Where we are now and where we are going to be,” arXiv preprint arXiv:1703.09809, 2017.

[36] P. Naghizadeh and M. Liu, “Provision of public goods on networks: On existence, uniqueness, and centralities,” IEEE Trans. Netw. Sci. Eng., vol. 5, no. 3, pp. 225–236, Jul.-Sep. 2017.

[37] F. Hu, Q. Hao, and K. Bao, “A survey on software-defined network and openflow: From concept to implementation,” IEEE Commun. Surveys Tuts., vol. 16, no. 4, pp. 2181–2206, Oct.-Dec. 2014.

[38] P. Dely, A. Kassler, and N. Bayer, “OpenFlow for wireless mesh networks,” in Proc. Int. Conf. Comput. Commun. Netw., Jul./Aug. 2011, pp. 1–6.

[39] P. K. Sharma, R. Mahajan, and Surender, ``A security architecture for attacks detection and authentication in wireless mesh networks," Cluster Comput., vol. 20, no. 3, pp. 2323_2332, Sep. 2017

[40] R. Vignesh et al., “Location monitoring system for maritime security using RSSI technology,” Int. Res. J. Eng. Technol., vol. 5, no. 4, pp. 2716–2718, 2018

[41] S. Srivastava and A. Srivastava, ``Integration of RSA and waterfall framework: Aggrandize security in cloud computing using integration of Rivest_Shamir_Adleman (encryption algorithm) and waterfall model," J. Microcontroller Eng. Appl., vol. 4, no. 3, pp. 1_8, 2018.

[42] V. Sivaraman, H. H. Gharakheili, C. Fernandes, N. Clark and T. Karliychuk, “Smart IoT Devices in the Home: Security and Privacy Implications,” IEEE Technol. Soc. Mag., vol. 37, no. 2, pp. 71-79, June 2018.

[43] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 2347-2376, June 2015.

[44] D. Tapscott and A. Tapscott, “Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World.”, Baltimore, MD, USA, Penguin, 2016.

[45] R. Rodrigo, P. Najera and J. Lopez, “Securing the Internet of Things (IoT)”, in IEEE Computer, vol. 44, pp. 51-58, 2011, DOI:10.1109/MC.2011.291.

[46] J. Liu, J. Wan, D. Jia, Bi Zeng, Di Li, C. Hsu and H. Chen, “High-Efficiency Urban Traffic Management in Context-Aware Computing and 5G Communication”, IEEE Communications Magazine, vol. 55, no. 1, pp. 34-40, January 2017.

[47] L. Hu, M. Qiu, J. Song, M. S. Hossain and A. Ghoneim, “Software defined healthcare networks”, in IEEE Wireless Communications, vol. 22, no. 6, pp. 67-75, December 2015, doi: 10.1109/MWC.2015.7368826.

[48] K. Zheng, S. Ou, J. Alonso-Zarate, M. Dohler, F. Liu and H. Zhu, “Challenges of massive access in highly dense LTE-advanced networks with machine-to-machine communications”, in IEEE Wireless Communications, vol. 21, no. 3, pp. 12-18, June 2014, doi: 10.1109/MWC.2014.6845044.

[49] G. Xiong, P. Sun, Y. Hu, J. Lan, and K. Li, “An optimized deployment mechanism for virtual middleboxes in NFV-and SDN-enabling network,” KSII Trans. Internet Inf. Syst., vol. 10, no. 8, pp. 3474–3497, Aug. 2016.

[50] A. K. Roy and A. K. Khan, ‘‘Architectural and security prospective of wireless mesh network,’’ Int. J. Comput. Intell. IoT, vol. 2, no. 1, pp. 1–5, 2019.

[51] .J. Kaye, J. Vertesi, J. Ferreira, B. Brown and M. Perry, “CHIMoney: Financial interactions, digital cash, capital exchange and mobile money”, in Proceeding Extended Abstr. Human Factors Computing System, pp. 111-114, April 2014.

[52] J. Lewandowska. [online] Available at : Fraud Resilient System for Off-Line Micro-Payments | Point Of Sale | Network Security (scribd.com). Accessed 10th January 2021.

[53] . J. Tellez Isaac and Z. Sherali, “Secure Mobile Payment Systems”, in IT Professional, vol. 16, no. 3, pp. 36-43, 2014. DOI: 10.1109/MITP.2014.40

[54] “TOTP: Time-based one-time password algorithm,” IETF, Fremont, CA, USA, RFC 6238, 2011.

[55] Aziz A, Singh K, Elsawy A, Osamy W, Khedr AM. "GWRA: grey wolf based reconstruction algorithm for compressive sensing signals," 2019. PeerJ Comput. Sci. 5:e217 DOI 10.7717/peerj-cs.217.

[56] V. Erceg et al., "An empirically based path loss model for wireless channels in suburban environments," in IEEE Journal on Selected Areas in Communications, vol. 17, no. 7, pp. 1205-1211, July 1999, doi: 10.1109/49.778178.

[57] D. Reuver, E. Verschuur, F. Nikayin, N. Cerpa and H. Bouwman, “Collective action for mobile payment platforms: A case study on collaboration issues between banks and telecom

operators”, Electron. Commerce Res. Appl., vol. 14, no. 5, pp. 331-344, September/October 2015.

[58] J. J. Liu, H. R. Li, Y. Gao, H. Yu and D. Jiang, “A Geohash-based index for spatial data management in distributed memory”, In Proceedings of the 22nd International Conference on Geoinformatics, pp. 1-4, 2014.

[59] S. Liu, K. Liu, J. Zhang, T. Zhang, Z. Xu, and F. Liu, ‘‘A location and cluster-based MAC and routing protocol for wireless mesh networks,’’ in Proc. 2nd IEEE Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC), May 2018, pp. 1963–1969.

[60] W. Mao and C. Boyd, “Towards formal analysis of security protocols,” in Proc. Comput. Found. Workshop VI, Franconia, NH, USA, 1993, pp. 147–158.

[61] B. Cyr, J. Mahmod and U. Guin, "Low-Cost and Secure Firmware Obfuscation Method for Protecting Electronic Systems From Cloning," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3700-3711, April 2019, doi: 10.1109/JIOT.2018.2890277.

[62] J. Y. Koh, I. Nevat, D. Leong and W. -C. Wong, "Geo-Spatial Location Spoofing Detection for Internet of Things," in IEEE Internet of Things Journal, vol. 3, no. 6, pp. 971-978, Dec. 2016, doi: 10.1109/JIOT.2016.2535165.

[63] M. Paliwal and D. Shrimankar, "Effective Resource Management in SDN Enabled Data Center Network Based on Traffic Demand," in IEEE Access, vol. 7, pp. 69698-69706, 2019, doi: 10.1109/ACCESS.2019.2919348.

[64] A. Montazerolghaem, M. H. Yaghmaee and A. Leon-Garcia, "Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation," in IEEE Transactions on Green Communications and Networking, vol. 4, no. 3, pp. 873-889, Sept. 2020, doi: 10.1109/TGCN.2020.2982821.

[65] Y. -T. Han, I. -Y. Hwang, C. -C. Kim and H. -S. Park, "A New Attainable TCP Throughput Measurement Tool for Long Distance High Speed Networks," in IEEE Communications Letters, vol. 14, no. 10, pp. 990-992, October 2010, doi: 10.1109/LCOMM.2010.091010.100646.

[66] Andonov, Ivailo & Tsvetanov, Simeon & Dimitrov, Stefan. (2019). “Securing IoT devices against cloning”.

[67] G. Akilarasu and S. M. Shalinie, ``Wormhole-free routing and DoS attack defense in wireless mesh networks,'' Wireless Netw., vol. 23, no. 6, pp. 1709_1718, Aug. 2017.

[68] R. K. Kodali and B. Kirti, "NS-3 Model of an IoT network," 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2020, pp. 699-702, doi: 10.1109/ICCCA49541.2020.9250808.

[69] The MEMSIC Solution: MICAz Mote Platform Datasheet. Accessed: Jun. 1, 2022. [Online]. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf

[70] R. Al Alawi, "RSSI based location estimation in wireless sensors networks," in Proc. IEEE Int. Conf. Netw., Singapore, 2011, pp. 118–122.

[71] M. H. Kutner et al., Applied Linear Regression Models, 4th ed. New York, NY, USA: McGraw-Hill, 2004.

[72] Radio Propagation Modeling, http://morse.colorado.edu/~tlen5510/text/classwebch3.html, [Online], Accessed in 30th August 2021.

[73] K. B. Frikken et al., "Robust authentication using physically unclonable functions," in Proc. Inf. Security Conf., Pisa, Italy, 2009, pp.262–277.

[74] S. Kumari et al., "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," J. Supercomput., vol. 74, no. 12, pp. 6428–6453, 2018.

[75] J. Mo, Z. Hu, and Y. Lin, "Remote user authentication and key agreement for mobile client–server environments on elliptic curve cryptography,"J. Supercomput., vol. 74, no. 11, pp. 5927–5943, Jul. 2018.

[76] X. Liu et al., "A Security Framework for the Internet of Things in the Future Internet Architecture," Future Internet, vol. 9, no. 3, p. 27, Jun. 2017.

[77] M.B. Frederick, "Cellular telephone anti-fraud system," U.S. Patent No. 5,448,760, Sept. 5, 1995.

[78] K. D. Hawkes, "Transient analysis system for characterizing RF transmitters by analyzing transmitted RF signals," U.S. Patent No. 5,758,277, May 26, 1998.

[79] J. Toonstra and W. Kinsner, "A radio transmitter fingerprinting system ODO-1," in Proc. IEEE Can. Conf. Elect. Comput. Eng., Calgary, Alta., 1996, pp. 60–63.

[80] J O.H. Tekbas, N. Serinken, and O. Ureten, “An experimental performance evaluation of a novel transmitter identification system under varying environmental conditions,” Can. J. Elect. Comput. Eng., vol. 29, no. 3, July 2004, pp. 203–209.

[81] J. Garay, A. Kiayias, and N. Leonardos, “The bitcoin backbone protocol: Analysis and applications,” in Advances in Cryptology – EUROCRYPT 2015, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 281–310.

[82] S. M. Glass, Vallipuram Muthukkumarasamy, Marius Portmann, “Detecting Man-in-the-Middle and Wormhole Attacks in Wireless Mesh Networks”, 2009.

[83] X. Liu et al., “A Security Framework for the Internet of Things in the Future Internet Architecture,” Futur. Internet, vol. 9, no. 3, p. 27, Jun. 2017.

[84] W. C. Suski II, M.A. Temple, M. J. Mendenhall, and R. F. Mills,“Radio frequency fingerprinting commercial communication devices to enhance electronic security,” International Journal of Electronic Security and Digital Forensics, vol. 1, no. 3, pp. 301–322, 2008.

[85] J. Zaddach, A. Costin, “Embedded Devices Security and Firmware Reverse Engineering”, 2013.

[86] A. Khalil, N. Mbarek and O. Togni, "Fuzzy Logic Based Security Trust Evaluation for IoT Environments," 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 2019, pp. 1-8, doi: 10.1109/AICCSA47632.2019.9035294.

[87] N. C. Kiran and G. N. Kumar, "Building robust m-commerce payment system on offline wireless network," 2011 Fifth IEEE International Conference on Advanced Telecommunication Systems and Networks (ANTS), 2011, pp. 1-3, doi: 10.1109/ANTS.2011.6163664.

[88] I. O. Kennedy, P. Scanlon, and M. M. Buddhikot, “Passive Steady State RF Fingerprinting: A Cognitive Technique for Scalable Deployment of Co-Channel Femto Cell Underlays,” in 2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2008, pp. 1–12.

[89] V. Osmov, A. Kurbanniyazov, R. Hussain, A. Oracevic, S. M. A. Kazmi and F. Hussain, "On the Blockchain-Based General-Purpose Public Key Infrastructure," 2019

[90] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, “IoTSense: Behavioral Fingerprinting of IoT Devices”.

[91] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, “Wireless device identification with radiometric signatures,” in Proceedings of the 14th ACM international conference on Mobile computing and networking - MobiCom ’08, 2008, p. 116.

[92] H. Patel, M. A. Temple, and B. W. Ramsey, “Comparison of High-end and Low-end Receivers for RF-DNA Fingerprinting,” in 2014 IEEE Military Communications Conference, 2014, pp. 24–29.

[93] J. François et al., “Automated Behavioral Fingerprinting,” Lect. Notes Comput. Sci., vol. 5758, 2009.

[94] SA. Haniz, K. Sano, R. Iwata, R. Kosaka, Y. Kuki, and G. Khanh TRAN, “A Guide of Fingerprint Based Radio Emitter Localization using Multiple Sensors.”.

[95] J. Yu, V. Cheval, and M. Ryan, “Dtki: A new formalized pki with verifiable trusted parties,” The Computer Journal, vol. 59, no. 11, pp. 1695–1713, Nov 2016.

[96] G. Liu, Q. Yang, H. Wang and A. X. Liu, "Trust Assessment in Online Social Networks," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 2, pp. 994-1007, 1 March-April 2021, doi: 10.1109/TDSC.2019.2916366.

[97] P. P. Ray, B. Chowhan, N. Kumar and A. Almogren, "BIoTHR: Electronic Health Record Servicing Scheme in IoT-Blockchain Ecosystem," in IEEE Internet of Things Journal, vol. 8, no. 13, pp. 10857-10872, 1 July1, 2021, doi: 10.1109/JIOT.2021.3050703.

[98] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, “Authentication in Distributed Systems: Theory and Practice,” In ACM Transactions on Computer Systems, Vol. 10, No. 4, 265–310, November 1992.

[99] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons", IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 466-490, 1st Quart. 2016.

[100] P. E. Numan, H. Park, C. Laoudias, S. Horsmanheimo and S. Kim, "DNN-based indoor fingerprinting localization with WiFi FTM", Proc. 23rd IEEE Int. Conf. Mobile Data Manage. (MDM), pp. 367-371, Jun. 2022.

[101] J. He and H. C. So, "A hybrid TDOA-fingerprinting-based localization system for LTE network", IEEE Sensors J., vol. 20, no. 22, pp. 13653-13665, Nov. 2020.

[102] H. Wu, X. Li, W. Dai, W. Zhao, “Mobile Payment Framework Based on 3G Network, Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops (ISECS ’10) Guangzhou, P. R. China, 29-31, pp. 172-175, July 2010.

[103] G. Baldini, G. Steri, R. Giuliani, and C. Gentile, “Imaging time series for internet of things radio frequency fingerprinting,” in 2017 International Carnahan Conference on Security Technology (ICCST), 2017, pp. 1–6.

[104] M. Miettinen, A.-R. Sadeghi, S. Marchal, N. Asokan, I. Hafeez, and S. Tarkoma, “IOT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT.”.

[105] C. T. Zenger, M. Pietersz, J. Zimmer, J.-F. Posielek, T. Lenze, and C. Paar, “Authenticated Key Establishment for Low-Resource Devices Exploiting Correlated Random Channels”.

[106] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, “GTID: A Technique for Physical Device & Device Type Fingerprinting,” IEEE Trans. Dependable Secur. Comput., vol. 12, no. 5, pp. 519– 532, Sep. 2015.

[107] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, “Identifying Wireless Users via Transmitter Imperfections,” IEEE J. Sel. Areas Commun., vol. 29, no. 7, pp. 1469–1479, Aug. 2011.

[108] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” Internet Requests for Comments, RFC Editor, RFC 5820, May 2008. [Online]. Available: https://tools.ietf.org/html/rfc5280

[109] B. Laurie, A. Langley, E. Kasper, E. Messeri, and R. Stradling, “Certificate Transparency Version 2.0,” Internet Requests for Comments, RFC Editor, RFC 6962, February 2019. [Online]. Available: https://tools.ietf.org/html/draft-ietf-trans-rfc6962-bis

[110] (2019) Repository of documentation and certificates. [Online]. Available: https://pki.goog/.

[111] (2018, August 10) Chrome certificate transparency requirements. [Online]. Available:https://fpki.idmanagement.gov/announcements/chromect/

[112] J. K. Millen and R. N. Wright, "Reasoning about trust and insurance in a public key infrastructure," Proceedings 13th IEEE Computer Security Foundations Workshop. CSFW-13, 2000, pp. 16-22, doi: 10.1109/CSFW.2000.856922.

[113] Y. Li, “Design of a key establishment protocol for smart home energy management system”, 2013.

[114] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, “Fingerprinting Wi-Fi Devices Using Software Defined Radios,” in Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks - WiSec ’16, 2016, pp. 3–14.

[115] D. R. Reising, M. A. Temple, and M. J. Mendenhall, “Improving Intra-Cellular Security Using Air Monitoring with RF Fingerprints,” in 2010 IEEE Wireless Communication and Networking Conference, 2010, pp. 1–6.

[116] B. Danev, D. Zanetti, and S. Capkun, “On physical-layer identification of wireless devices,” ACM Comput. Surv., vol. 45, no. 1, pp. 1–29, Nov. 2012.

[117] M. D. Williams, M. A. Temple, and D. R. Reising, “Augmenting Bit-Level Network Security Using Physical Layer RF-DNA Fingerprinting,” in 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, 2010, pp. 1–6.

[118] T. Suzuki and N. Kubo, “GNSS-SDRLIB: An open-source and real-time GNSS software defined radio library,” in Proc. of the 27th Intl. Technical Meeting of the Satellite Division of The Institute of Navigation, Tampa, FL, Sept. 2014, pp. 1364– 1375.

[119] D. R. Reising, M. A. Temple, and M. J. Mendenhall, “Improved wireless security for GMSK-based devices using RF fingerprinting,” Int. J. Electron. Secur. Digit. Forensics, vol. 3, no. 1, p. 41, 2010

[120] C. Ellison et al, “SPKI Certificate Theory,” IETF Network Working Group RFC 2693, September 1999. Posted at http://www.ietf.org/ rfc/rfc2693.txt.

[121] B. Boashash, “Estimating and interpreting the instantaneous frequency of a signal: Part I: Fundamentals,” Proc. IEEE, vol. 80, no. 4, Apr. 1992, pp. 520–538.

[122] Dang et al. designed an MKR encapsulated in PDMS with sensitivity of −8.48 nm C in 2020.

[123] C.M. Bishop, Neural Networks for Pattern Recognition, New York: Oxford University Press, 2004, pp. 295–319.

[124] D.F. Specht, “Probabilistic neural networks,” Neural Networks, vol. 3, no. 1, 1990, pp. 109–118.

[125] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, “Robust Threshold DSS Signatures,” In Advances in Cryptology—CRYPTO ’96, Lecture Notes in Computer Science 1070, 354–371, Springer-Verlag, 1996.

[126] R. Berta, F. Bellotti, A. De Gloria, L. Lazzaroni, “Assessing Versatility of a Generic End-to-End Platform for IoT Ecosystem Applications”. Sensors 2022, 22, 713. https://doi.org/10.3390/s22030713

[127] D. Boneh and M. Franklin, “Efficient generation of shared RSA keys,” In Advances in Cryptology— CRYPTO ’97, Lecture Notes in Computer Science 1294, 424–439, Springer-Verlag, 1997.

[128] S. Huh, S. Cho, S. Kim, “Managing IoT devices using blockchain platform”, 2017.

[129] X. Liu et al., “A Security Framework for the Internet of Things in the Future Internet Architecture,” Future Internet, vol. 9, no. 3, p. 27, Jun. 2017.

[130] S. Khandker, H. Turtiainen, A. Costin, and T. Hamalainen,``Cybersecurity attacks on software logic and error handling within ADS-B implementations: Systematic testing of resilience and countermeasures,' IEEE Trans. Aerosp. Electron. Syst., early access, Dec. 31, 2021, doi:10.1109/TAES.2021.3139559.

[131] M.B. Frederick, “Cellular telephone anti-fraud system,” U.S. Patent No. 5,448,760, Sept. 5, 1995.

[132] K. D. Hawkes, “Transient analysis system for characterizing RF transmitters by analyzing transmitted RF signals,” U.S. Patent No. 5,758,277, May 26, 1998.

[133] J. Toonstra and W. Kinsner, “A radio transmitter fingerprinting system ODO-1,” in Proc. IEEE Can. Conf. Elect. Comput. Eng., Calgary, Alta., 1996, pp. 60–63.

[134] J O.H. Tekbas, N. Serinken, and O. Ureten, “An experimental performance evaluation of a novel transmitter identification system under varying environmental conditions,” Can. J. Elect. Comput. Eng., vol. 29, no. 3, July 2004, pp. 203–209.

[135] J. Garay, A. Kiayias, and N. Leonardos, “The bitcoin backbone protocol: Analysis and applications,” in Advances in Cryptology – EUROCRYPT 2015, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 281–310.

[136] S. M. Glass, Vallipuram Muthukkumarasamy, Marius Portmann, “Detecting Man-in-the-Middle and Wormhole Attacks in Wireless Mesh Networks”, 2009.

[137] X. Liu et al., “A Security Framework for the Internet of Things in the Future Internet Architecture,” Futur. Internet, vol. 9, no. 3, p. 27, Jun. 2017.

[138] T. Wang et al., “An intelligent edge-computing-based method to counter coupling problems in cyber-physical systems,” IEEE Network., vol. 34, no. 3, pp. 16–22, May/Jun. 2020.

[139] W. C. Suski II, M.A. Temple, M. J. Mendenhall, and R. F. Mills,“Radio frequency fingerprinting commercial communication devices to enhance electronic security,” International Journal of Electronic Security and Digital Forensics, vol. 1, no. 3, pp. 301–322, 2008.

[140] A. Goudossis and S. K. Katsikas, ``Towards a secure automatic identification system (AIS),' J. Mar. Sci. Technol., vol. 24, no. 2, pp. 410_423, Jun. 2019.

[141] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, “Eyeriss V2: A flexible accelerator for emerging deep neural networks on mobile devices,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 9, no. 2, pp. 292–308, Jun. 2019.

[142] J. Zaddach, A. Costin, “Embedded Devices Security and Firmware Reverse Engineering”, 2013.

[143] A. Khalil, N. Mbarek and O. Togni, "Fuzzy Logic Based Security Trust Evaluation for IoT Environments," 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 2019, pp. 1-8, doi: 10.1109/AICCSA47632.2019.9035294.

[144] S. Colabianchi, F. Costantino, G. Di Gravio, F. Nonino, and R. Patriarca,“Discussing resilience in the context of cyber physical systems,” Comput. Ind.Eng., vol. 160, Oct. 2021, Art. no. 107534, doi: 10.1016/J.CIE.2021.107534.

[145] N. C. Kiran and G. N. Kumar, "Building robust m-commerce payment system on offline wireless network," 2011 Fifth IEEE International Conference on Advanced Telecommunication Systems and Networks (ANTS), 2011, pp. 1-3, doi: 10.1109/ANTS.2011.6163664.

[146] I. O. Kennedy, P. Scanlon, and M. M. Buddhikot, “Passive Steady State RF Fingerprinting: A Cognitive Technique for Scalable Deployment of Co-Channel Femto Cell Underlays,” in 2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2008, pp. 1–12.

[147] V. Osmov, A. Kurbanniyazov, R. Hussain, A. Oracevic, S. M. A. Kazmi and F. Hussain, "On the Blockchain-Based General-Purpose Public Key Infrastructure," 2019

[148] D. E. Culler and H. Mulder, “Smart sensors to network the world,” Sci. Amer.,vol. 290, no. 6, pp. 84–91, 2004, doi: 10.1038/SCIENTIFICAMERICAN0604-84.

[149] S. N. Vecherin, D. K. Wilson, and C. L. Pettit, “Optimal sensor placement withsignal propagation effects and inhomogeneous coverage preferences,”Int. J.Sensor Netw., vol. 9, no. 2, pp. 107–120, 2011, doi: 10.1504/IJSNET.2011.038763.

[150] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” in Proc. IEEE Int. Conf. Commun., 2001, vol. 2, pp. 472–476.

[151] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, “IoTSense: Behavioral Fingerprinting of IoT Devices”.

[152] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, “Wireless device identification with radiometric signatures,” in Proceedings of the 14th ACM international conference on Mobile computing and networking - MobiCom ’08, 2008, p. 116.

[153] H. Patel, M. A. Temple, and B. W. Ramsey, “Comparison of High-end and Low-end Receivers for RF-DNA Fingerprinting,” in 2014 IEEE Military Communications Conference, 2014, pp. 24–29.

[154] J. François et al., “Automated Behavioral Fingerprinting,” Lect. Notes Comput. Sci., vol. 5758, 2009.

[155] SA. Haniz, K. Sano, R. Iwata, R. Kosaka, Y. Kuki, and G. Khanh TRAN, “A Guide of Fingerprint Based Radio Emitter Localization using Multiple Sensors.”.

[156] J. Yu, V. Cheval, and M. Ryan, “Dtki: A new formalized pki with verifiable trusted parties,” The Computer Journal, vol. 59, no. 11, pp. 1695–1713, Nov 2016.

[157] G. Liu, Q. Yang, H. Wang and A. X. Liu, "Trust Assessment in Online Social Networks," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 2, pp. 994-1007, 1 March-April 2021, doi: 10.1109/TDSC.2019.2916366.

[158] P. P. Ray, B. Chowhan, N. Kumar and A. Almogren, "BIoTHR: Electronic Health Record Servicing Scheme in IoT-Blockchain Ecosystem," in IEEE Internet of Things Journal, vol. 8, no. 13, pp. 10857-10872, 1 July1, 2021, doi: 10.1109/JIOT.2021.3050703.

[159] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, “Authentication in Distributed Systems: Theory and Practice,” In ACM Transactions on Computer Systems, Vol. 10, No. 4, 265–310, November 1992.

[160] A. Alsufyani, Y. Alotaibi, A. O. Almagrabi, S. A. Alghamdi, and N. Alsufyani,“Optimized intelligent data management framework for a cyber-physical system for computational applications,” Complex Intell. Syst., vol. 1, pp. 1–13, Aug.2021, doi: 10.1007/S40747-021-00511-W.

[161] A. El Saddik, “Digital Twins: The convergence of multimedia technologies,”IEEE Multimedia, vol. 25, no. 2, pp. 87–92, Apr.–Jun. 2018.

[162] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,”IEEE Trans. Ind. Inform., vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[163] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons", IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 466-490, 1st Quart. 2016.

[164] P. E. Numan, H. Park, C. Laoudias, S. Horsmanheimo and S. Kim, "DNN-based indoor fingerprinting localization with WiFi FTM", Proc. 23rd IEEE Int. Conf. Mobile Data Manage. (MDM), pp. 367-371, Jun. 2022.

[165] J. He and H. C. So, "A hybrid TDOA-fingerprinting-based localization system for LTE network", IEEE Sensors J., vol. 20, no. 22, pp. 13653-13665, Nov. 2020.

[166] H. Wu, X. Li, W. Dai, W. Zhao, “Mobile Payment Framework Based on 3G Network, Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops (ISECS ’10) Guangzhou, P. R. China, 29-31, pp. 172-175, July 2010.

[167] G. Baldini, G. Steri, R. Giuliani, and C. Gentile, “Imaging time series for internet of things radio frequency fingerprinting,” in 2017 International Carnahan Conference on Security Technology (ICCST), 2017, pp. 1–6.

[168] M. Miettinen, A.-R. Sadeghi, S. Marchal, N. Asokan, I. Hafeez, and S. Tarkoma, “IOT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT.”.

[169] C. T. Zenger, M. Pietersz, J. Zimmer, J.-F. Posielek, T. Lenze, and C. Paar, “Authenticated Key Establishment for Low-Resource Devices Exploiting Correlated Random Channels”.

[170] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, “GTID: A Technique for Physical Device & Device Type Fingerprinting,” IEEE Trans. Dependable Secur. Comput., vol. 12, no. 5, pp. 519– 532, Sep. 2015.

[171] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, “Identifying Wireless Users via Transmitter Imperfections,” IEEE J. Sel. Areas Commun., vol. 29, no. 7, pp. 1469–1479, Aug. 2011.

[172] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” Internet Requests for Comments, RFC Editor, RFC 5820, May 2008. [Online]. Available: https://tools.ietf.org/html/rfc5280

[173] B. Laurie, A. Langley, E. Kasper, E. Messeri, and R. Stradling, “Certificate Transparency Version 2.0,” Internet Requests for Comments, RFC Editor, RFC 6962, February 2019. [Online]. Available: https://tools.ietf.org/html/draft-ietf-trans-rfc6962-bis

[174] (2019) Repository of documentation and certificates. [Online]. Available: https://pki.goog/.

[175] (2018, August 10) Chrome certificate transparency requirements. [Online]. Available:https://fpki.idmanagement.gov/announcements/chromect/

[176] J. K. Millen and R. N. Wright, "Reasoning about trust and insurance in a public key infrastructure," Proceedings 13th IEEE Computer Security Foundations Workshop. CSFW-13, 2000, pp. 16-22, doi: 10.1109/CSFW.2000.856922.

[177] Y. Li, “Design of a key establishment protocol for smart home energy management system”, 2013.

[178] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, “Fingerprinting Wi-Fi Devices Using Software Defined Radios,” in Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks - WiSec ’16, 2016, pp. 3–14.

[179] D. R. Reising, M. A. Temple, and M. J. Mendenhall, “Improving Intra-Cellular Security Using Air Monitoring with RF Fingerprints,” in 2010 IEEE Wireless Communication and Networking Conference, 2010, pp. 1–6.

[180] B. Danev, D. Zanetti, and S. Capkun, “On physical-layer identification of wireless devices,” ACM Comput. Surv., vol. 45, no. 1, pp. 1–29, Nov. 2012.

[181] M. D. Williams, M. A. Temple, and D. R. Reising, “Augmenting Bit-Level Network Security Using Physical Layer RF-DNA Fingerprinting,” in 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, 2010, pp. 1–6.

[182] T. Suzuki and N. Kubo, “GNSS-SDRLIB: An open-source and real-time GNSS software defined radio library,” in Proc. of the 27th Intl. Technical Meeting of the Satellite Division of The Institute of Navigation, Tampa, FL, Sept. 2014, pp. 1364– 1375.

[183] D. R. Reising, M. A. Temple, and M. J. Mendenhall, “Improved wireless security for GMSK-based devices using RF fingerprinting,” Int. J. Electron. Secur. Digit. Forensics, vol. 3, no. 1, p. 41, 2010

[184] C. Ellison et al, “SPKI Certificate Theory,” IETF Network Working Group RFC 2693, September 1999. Posted at http://www.ietf.org/ rfc/rfc2693.txt.

[185] B. Boashash, “Estimating and interpreting the instantaneous frequency of a signal: Part I: Fundamentals,” Proc. IEEE, vol. 80, no. 4, Apr. 1992, pp. 520–538.

[186] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, “Robust Threshold DSS Signatures,” In Advances in Cryptology—CRYPTO ’96, Lecture Notes in Computer Science 1070, 354–371, Springer-Verlag, 1996.

[187] R. Berta, F. Bellotti, A. De Gloria, L. Lazzaroni, “Assessing Versatility of a Generic End-to-End Platform for IoT Ecosystem Applications”. Sensors 2022, 22, 713. https://doi.org/10.3390/s22030713

[188] D. Boneh and M. Franklin, “Efficient generation of shared RSA keys,” In Advances in Cryptology— CRYPTO ’97, Lecture Notes in Computer Science 1294, 424–439, Springer-Verlag, 1997.

[189] S. Huh, S. Cho, S. Kim, “Managing IoT devices using blockchain platform”, 2017.

[190] J. G. V. Etibou and S. Pierre, "IoT Devices Modular Security Approach Using Positioning Security Engine," in IEEE Access, vol. 12, pp. 147659-147670, 2024, doi: 10.1109/ACCESS.2024.3424658.

APPENDIX A GNUPLOT Execution Protocol Articles 2 And 3

Step 1:

Obtain a ARduino MKR board with GSM module for Interaction with users.

Step 2:

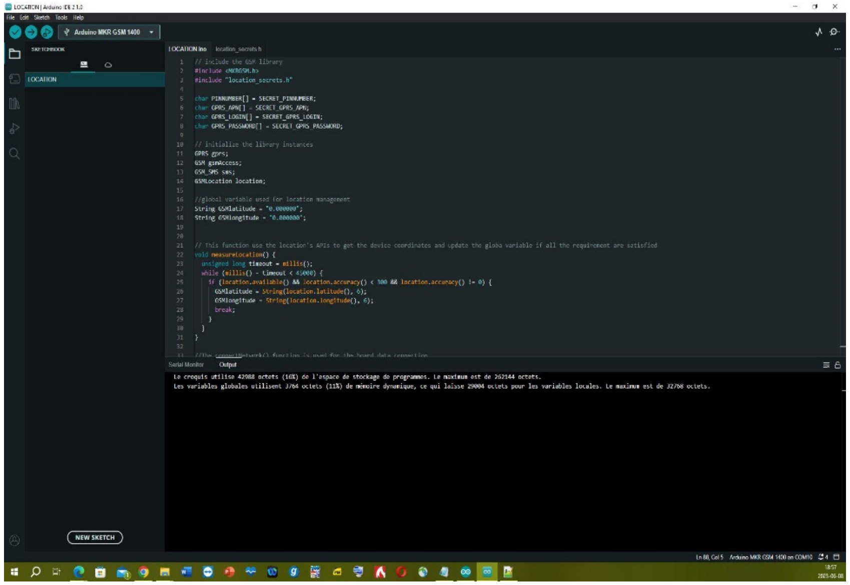
Connect the Board and run the code MKR_GSM_SMS_Localize_20230625173648. And save the Log.

Step 3:

Store de data log in separate data plot file directory. Run de code mygnuplot.plt from your internal storage using GNU PLOT.

APPENDIX B Arduino MKR 1400 System IDE Article 2 and 3

Mkr board 1400 with SIM card



```
// include the GSM library
#include <MKRGSM.h>
#include "location_secrets.h"

char PINNUMBER[] = SECRET_PINNUMBER;
char GPRS_APN[] = SECRET_GPRS_APN;
char GPRS_LOGIN[] = SECRET_GPRS_LOGIN;
char GPRS_PASSWORD[] = SECRET_GPRS_PASSWORD;

// initialize the library instances
GPRS gprs;
GSM gsmAccess;
GSM_SMS sms;
GSMLocation location;

//global variable used for location management
String GSMlatitude = "0.000000";
String GSMlongitude = "0.000000";


// This function use the location's APIs to get the device coordinates and update the globa variable if all the requirement are satisfied
void measureLocation() {
  unsigned long timeout = millis();
  while (millis() - timeout < 45000) {
    if (location.available() && location.accuracy() < 300 && location.accuracy() != 0) {
      GSMlatitude = String(location.latitude(), 6);
      GSMlongitude = String(location.longitude(), 6);
      break;
    }
  }
}

//The connectNetwork() function is used for the board data connection
```

```
Le croquis utilise 42988 octets (16%) de l'espace de stockage de programmes. Le maximum est de 262144 octets.
Les variables globales utilisent 3764 octets (11%) de mémoire dynamique, ce qui laisse 29004 octets pour les variables locales. Le maximum est de 32768 octets.
```

Arduino Main SE Code uploaded to MKR board