IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 31 March 2024, accepted 16 April 2024, date of publication 8 July 2024, date of current version 17 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3424658

RESEARCH ARTICLE

IoT Devices Modular Security Approach Using Positioning Security Engine

JEAN G. V. ETIBOU, (Student Member, IEEE), AND SAMUEL PIERRE, (Senior Member, IEEE)
Department of Computer and Software Engineering, Polytechnique Montréal, Montreal, QC H3T 1J4, Canada
Corresponding author: Jean G. V. Etibou (jean-gerald-vincent.etibou@polymtl.ca)

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, Smart city 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 transaction in an IoT environment. Therefore, to compensate 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 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 and extended secure capable environment for carrying 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.

I. 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 [1].

Wireless networks, especially ad hoc networks, benefit from device-to-device transmission signal 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 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 [2], [3], and safety and security of very high frequency (VHF) radio networks, transmitter identification system [4], [5]. Securing wireless channel 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 [6].

The security of an IoT network can be enhanced if users can confirm their identity and if the RF (Radio Frequency)

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

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 [7]. Although, IoT networks benefit from accessibility, flexibility and usability while being exposed to privacy and controllability concern [1], 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 [8].

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 [9] 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 [10].

The originality of this paper is that we consider offering a secure security engine to support IoT devices in IoT environment since IoT device has 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 [1]. Our design of the 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 [23]. 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 [11]. 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 II presents background and related work. Section III describes the proposed security engine system. Tests, results, and discussions are detailed in section IV. Finally, section V concludes this paper.

II. BACKGROUND AND RELATED WORK

Most IoT receiver used in RF fingerprinting techniques are large and expensive [11]. 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 [12].

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 [13].

A. BACKGROUND

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

1) 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 [14].

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 [15]. 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 [16].

Moreover, radio transmitters offer inherent nonlinearities that can be analyzed, then extracted as RF fingerprints of the signals [17].

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.

2) SECURITY ENGINE (SE)

Secure engine (SE) communication makes use of Public Key Infrastructure (PKI) defines as an asymmetric key-based security infrastructure [18]. 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 a 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 own 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 [19]. 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.

3) 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) [20] and GNU Privacy Guard (GnuPG) [21].

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.

4) INSURANCE AND TRUST IN SIMPLE PUBLIC-KEY INFRASTRUCTURE (SPKI)

The insurance logic is descried 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. [22] 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 be accomplished [10]. The user of the system is called 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 insurer's 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 into the transaction by assuming that the latter entities is trustworthy than a key signed by those same entities.

TABLE 1. Abreviations.

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 X_W with m amount (insurance certificate)
$(A) \rightarrow (B)$	"*says*" relation between entities A and B
HE	Hybrid Element
NWE	Network Element
RSE	Recipient Secure Element
SSE	Sender Secure Element
PvK	Private Key
PbK	Public Key

Nonetheless, there is no absolute guarantee that those trustworthy entities could still be unscrupulous certificates' insurers. Table 1 illustrates the insurance logic annotation used in this paper.

B. RELATED WORK

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

1) 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 and Chan [23] 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. [24] 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 and So [25] 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. [26] 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. [27] 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 time and will consume a lot of resources to process all the data information for fingerprinting purpose only.

Based of network traffic analysis, Miettinen et al. [28] 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 are very minim 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 [29]. Radhakrishna et al. [30] proposed a mechanism based on location channel randomness pairing. The work has been tested only on their prototype and required an implementation on all devices.

2) 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 [31] 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. [32] 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.

The attempt to propose a solution by Laurie et al. [33] 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 [34]. Moreover, this approach adds transparency to the current PKI architecture but do not guarantee the existence of illegitimate forged certificates in the certificate vault logs [35].

In the context of IoT networks, authors in [36] 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.

Li [37] 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 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 serves as an upgrade to an existing Simple Public Key Infrastructure (SPKI) certificate system [38].

III. PROPOSED SYSTEM

The proposed security system combines Enhanced RF fingerprinting (ERFF) by 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).

A. SYSTEM OVERVIEW

The proposed system architecture, as shown in Figure 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 for gaining access to UI's application requested services.

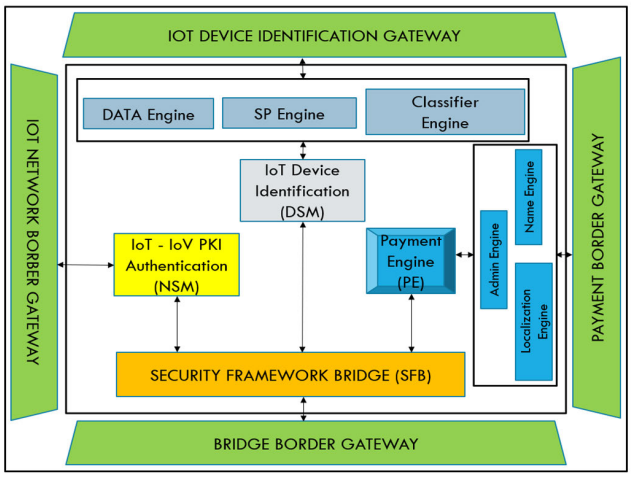

FIGURE 1. Proposed security engine infrastructure SE.

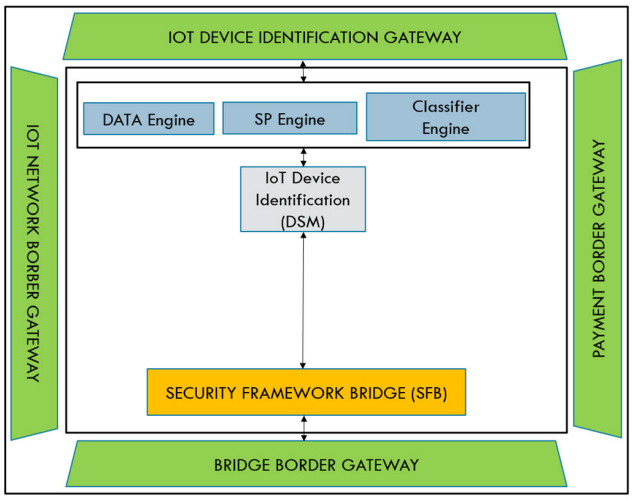

FIGURE 2. Device security module (DSM) system module of SE.

1) 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 [39]. 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 [40]. DSM performs data acquisition in either active or passive mode [41].

RF features extraction [42] 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 [43] 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 classifier follows an approach initially called Bayesian [44] 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 2.

2) 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 3.

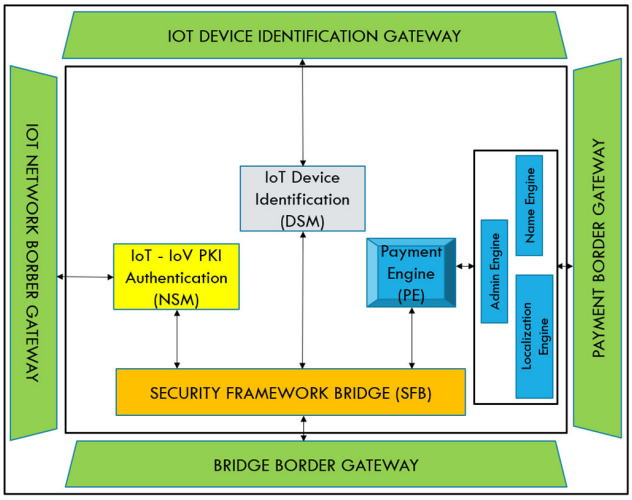

FIGURE 3. Network security module (NSM) system module of SE.

3) SECURITY FRAMEWORK BRIDGE (SFB)

The Security Framework Bridge (SFB) functions as Trust Third Party (TTP). IoT devices requests for a digital certificate through forwarded request by NSM. SFB includes two on demand functions that are essentials 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.

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 fasten the processing time for the transactions and provide the additional resources that might 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 key to UI for decryption.

In our proposed SE system, SFB acts as bridge to DSM, NSM, PE and UI, as shown in Figure 4.

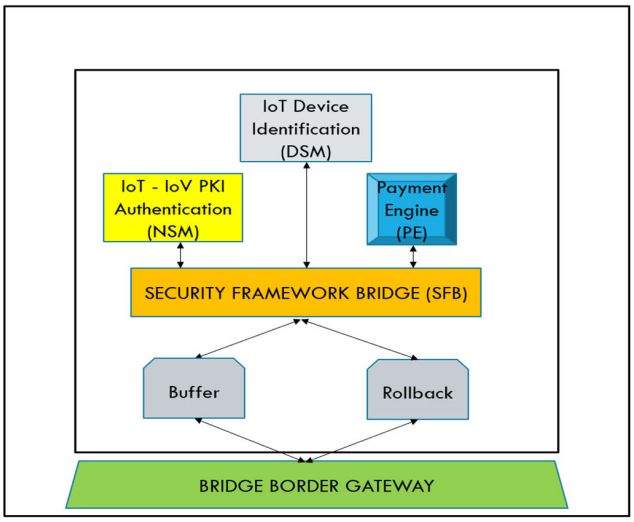

FIGURE 4. Security framework bridge (SFB) system module of SE.

B. SYSTEM WORKFLOWS

1) 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 5 follows 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 [45]. It is worth noting that the NSM module in Figure 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.

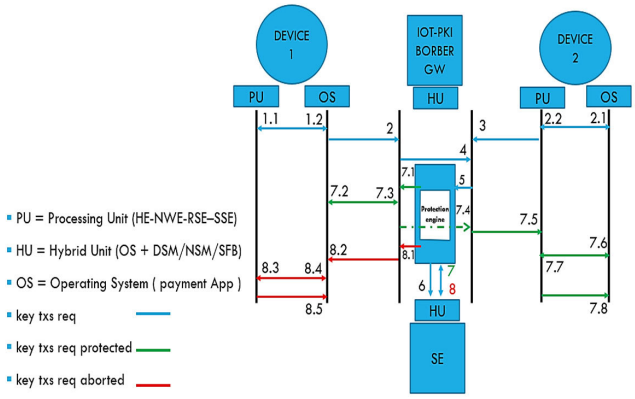

FIGURE 5. Proposed certificates and keys exchange scheme.

(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 generate 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.

2) INSURANCE AND TRUST SCHEME

Insurance and trust scheme refers to authentication in large-scale distributed systems according to LABW logic [22], interprets a certificate as a statement representation. There are two paths to building insurance: the shorter path and the longer path.

a: SHORTEST PATH

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

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

(2) $K_{D_3} \Rightarrow D_3$
(1) is interpreted as
(3) K_{D_2} says $K_{D_1} \Rightarrow D_1$ AND K_{D_3} says $K_{D_2}\$D_3$
(1) AND (3) give
(4) D_3 says $K_{D_2}\$D_3$
(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 D_3 has insured the link relation between D_1 and K_{D_1}.

Moreover, because (6) may not apply since:
(7) $K_{D_1} \Rightarrow D_1$ is true.

b: 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_1says$(K_{D_4} \Rightarrow D_4)$ then $K_{D_4} \Rightarrow D_4$
(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 K_{D_4} is damaged by any user device, then
(5) is false

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

3) SECURITY ENGINE'S PROPOSED ALGORITHM

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 1 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. The 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).

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 6 with the SE core payment modules components.

Algorithm 1 Security Engine Core Algorithm

Input: S(V).
$V = \{M \cup K\}$,
$M = \{DSM,NSM,SFB,PE\}$,
$K_{(n,m)} = \{PvIDK_1, PbIDK_2, PvIDK_n \ldots PbIDK_m\}$.
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.

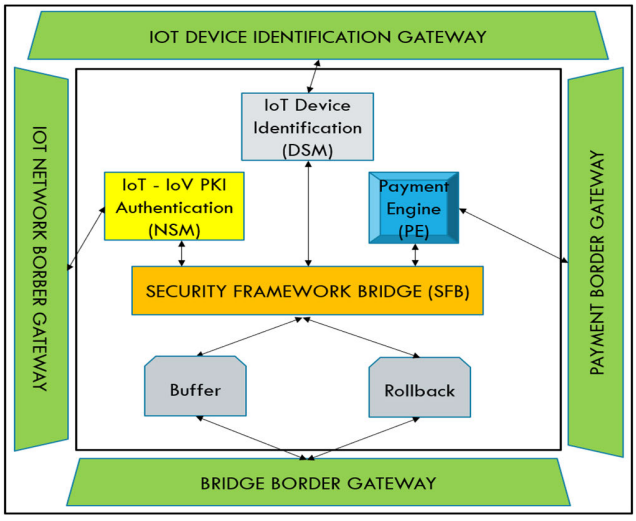

FIGURE 6. Security framework bridge (SFB) system with payment protection.

4) 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

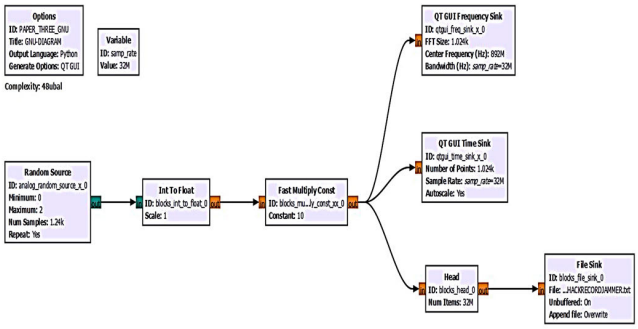

FIGURE 7. Proposed SE GNU representation block diagram.

phone calls, mobile date payment, and sending MMS and SMS, as shown in Figure 7.

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 being compatible with the arduino test experimental board GSM MKR [49], as illustrated in Table 2.

TABLE 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				

IV. RESULTS AND DISCUSSION

A. EXPERIMENTAL SETUP

For this experiment, Table 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 [50].

TABLE 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 the benefits of the simulation, we use Software Defined Radio (SDR) software GNU Radio [51] as a signal processing engine while the device hardware provides the RF front end. GNU Radio provides and 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 [49].

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 [50].

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.

In our current experimental setup, we made the abstraction of possible generation of audited key [51], [52] to privilege faster computation and keep this latter within the circumscription of the proposed Security Engine (SE).

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

The Table 4 illustrates IoT device interaction with IoT user application and device authentication results.

The Table 5 represents PKI key exchange results.

The Table 6 represents the performance metrics for the proposed SE system.

C. DISCUSSION OF RESULTS

Our SE's features, tested on the arduino MKR 1400 are compared against IoT 1 and IoT 2 devices in Table 7. Unlike IoT 1 and IoT 2, our SE does not require additional device companion to be able to offers user device interaction and SSL authentication. In Table 6, DSM phase 1 for instance, the subtraction between the start timestamp and the stop

TABLE 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

TABLE 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

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 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 and Table 5.

Our approach to device identification makes the use of the GSM module built into the MKR board. This latter uses the GSM library to send and receive SMS in 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 8.

Our approach to authenticate the users using SSL management of the MKR board brings SSL client 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 is supporting TLS/SSL for MQTTS client, 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

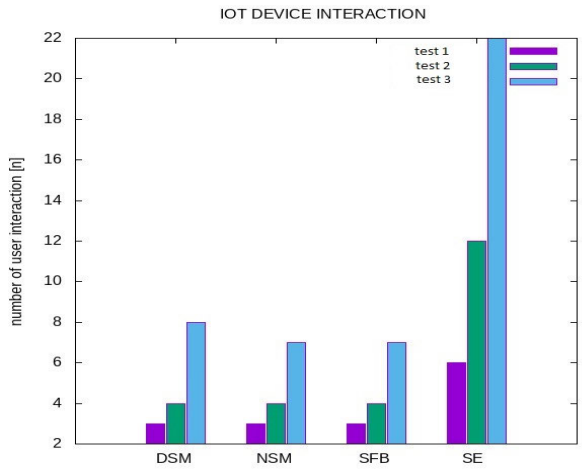

FIGURE 8. Number of interactions with SE.

connects in 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 to find that 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 SE system secured for any user client IoT connected devices or applications, as shown in Figure 9.

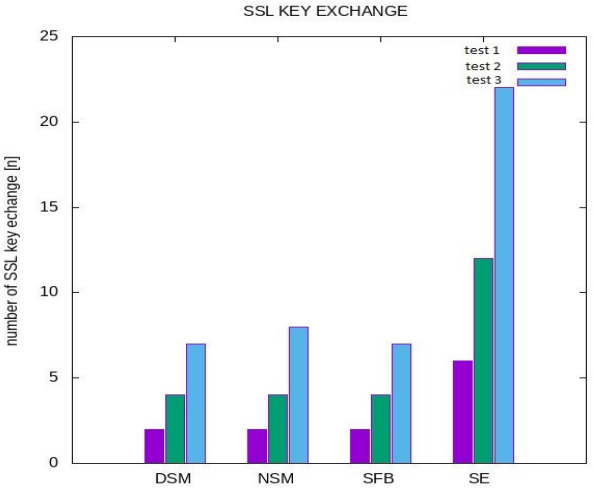

FIGURE 9. Number of SSL key exchange with SE.

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.

TABLE 6. SE performance metrics.

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

TABLE 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

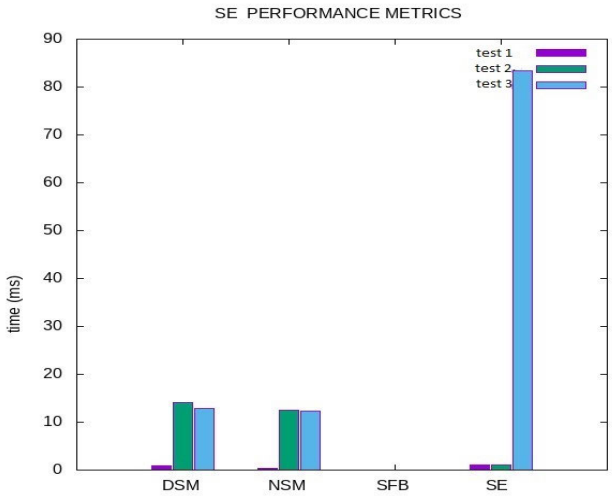

FIGURE 10. SE response time performance metrics.

Figure 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.

V. 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 using with responses from 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 limitation 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 article.

REFERENCES

[1] X. Liu, M. Zhao, S. Li, F. Zhang, and W. Trappe, "A security framework for the Internet of Things in the future internet architecture," *Future Internet*, vol. 9, no. 3, p. 27, Jun. 2017.

[2] M.B. Frederick, "Cellular telephone anti-fraud system," U.S. Patent 5 448 760, Sep. 5, 1995.

[3] K. D. Hawkes, "Transient analysis system for characterizing RF transmitters by analyzing transmitted RF signals," U.S. Patent 5 758 277, May 26, 1998.

[4] J. Toonstra and W. Kinsner, "A radio transmitter fingerprinting system ODO-1," in *Proc. Can. Conf. Electr. Comput. Eng.*, May 1996, pp. 60–63.

[5] O. H. Tekbas, N. Serinken, and O. Ureten, "An experimental performance evaluation of a novel radio-transmitter identification system under diverse environmental conditions," *Can. J. Electr. Comput. Eng.*, vol. 29, no. 3, pp. 203–209, Jul. 2004.

[6] 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, Germany: Springer, 2015, pp. 281–310.

[7] S. M. Glass, V. Muthukkumarasamy, and M. Portmann, "Detecting man-in-the-middle and wormhole attacks in wireless mesh networks," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.*, May 2009, pp. 530–538.

[8] W. C. S. Ii, M. A. Temple, M. J. Mendenhall, and R. F. Mills, "Radio frequency fingerprinting commercial communication devices to enhance electronic security," *Int. J. Electron. Secur. Digit. Forensics*, vol. 1, no. 3, p. 301, 2008.

[9] J. Zaddach and A. Costin, "Embedded devices security and firmware reverse engineering," Tech. Rep., 2013.

[10] A. Khalil, N. Mbarek, and O. Togni, "Fuzzy logic based security trust evaluation for IoT environments," in *Proc. IEEE/ACS 16th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Abu Dhabi, United Arab Emirates, Nov. 2019, pp. 1–8, doi: 10.1109/aiccsa47632.2019.9035294.

[11] N. C. Kiran and G. N. Kumar, "Building robust m-commerce payment system on offline wireless network," in *Proc. 5th IEEE Int. Conf. Adv. Telecommun. Syst. Netw. (ANTS)*, Dec. 2011, pp. 1–3, doi: 10.1109/ANTS.2011.6163664.

[12] 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 *Proc. 3rd IEEE Symp. New Frontiers Dyn. Spectr. Access Netw.*, Oct. 2008, pp. 1–12.

[13] V. Osmov, A. Kurbanniyazov, R. Hussain, A. Oracevic, S. M. A. Kazmi, and F. Hussain, "On the blockchain-based general-purpose public key infrastructure," in *Proc. IEEE/ACS 16th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2019.

[14] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "IoTSense: Behavioral fingerprinting of IoT devices," Tech. Rep.

[15] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, Sep. 2008, p. 116.

[16] H. Patel, M. A. Temple, and B. W. Ramsey, "Comparison of high-end and low-end receivers for RF-DNA fingerprinting," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2014, pp. 24–29.

[17] J. François, "Automated Behavioral Fingerprinting," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, vol. 5758, 2009, pp. 182–201.

[18] S. Haniz, K. Sano, R. Iwata, R. Kosaka, Y. Kuki, and G. K. Tran, "A guide of fingerprint based radio emitter localization using multiple sensors,"*IEICE Trans. Commun.*, vol. 101, no. 10, pp. 2104–2119, 2018.

[19] J. Yu, V. Cheval, and M. Ryan, "DTKI: A new formalized PKI with verifiable trusted parties," *Comput. J.*, vol. 59, no. 11, pp. 1695–1713, Nov. 2016.

[20] G. Liu, Q. Yang, H. Wang, and A. X. Liu, "Trust assessment in online social networks," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 994–1007, Mar. 2021, doi: 10.1109/TDSC.2019.2916366.

[21] P. P. Ray, B. Chowhan, N. Kumar, and A. Almogren, "BIoTHR: Electronic health record servicing scheme in IoT-blockchain ecosystem," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10857–10872, Jul. 2021, doi: 10.1109/JIOT.2021.3050703.

[22] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in distributed systems: Theory and practice," *ACM Trans. Comput. Syst.*, vol. 10, no. 4, pp. 265–310, Nov. 1992.

[23] 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.

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

[25] 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.

[26] H. Wu, X. Li, W. Dai, and W. Zhao, "Mobile payment framework based on 3G network," in *Proc. 3rd Int. Symp. Electron. Commerce Secur. Workshops (ISECS)*, Guangzhou, China, Jul. 2010, pp. 172–175.

[27] G. Baldini, G. Steri, R. Giuliani, and C. Gentile, "Imaging time series for Internet of Things radio frequency fingerprinting," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2017, pp. 1–6.

[28] 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," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 2177–2184.

[29] 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," *Comput. Netw.*, vol. 109, pp. 105–123, Nov. 2016.

[30] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, "GTID: A technique for physical device and device type fingerprinting," *IEEE Trans. Depend. Secure Comput.*, vol. 12, no. 5, pp. 519–532, Sep. 2015.

[31] 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.

[32] 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*, document RFC 5820, RFC Editor, Internet Requests for Comments, May 2008. [Online]. Available: https://tools.ietf.org/html/rfc5280

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

[34] (2019) *Repository of Documentation and Certificates*. [Online]. Available: https://pki.goog/

[35] (Aug. 10, 2018). *Chrome Certificate Transparency Requirements*. [Online]. Available: https://fpki.idmanagement.gov/announcements/chromect/

[36] J. K. Millen and R. N. Wright, "Reasoning about trust and insurance in a public key infrastructure," in *Proc. 13th IEEE Comput. Secur. Found. Workshop (CSFW)*, Aug. 2000, pp. 16–22, doi: 10.1109/CSFW.2000.856922.

[37] Y. Li, "Design of a key establishment protocol for smart home energy management system," in *Proc. 5th Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jun. 2013, pp. 88–93.

[38] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Fingerprinting Wi-Fi devices using software defined radios," in *Proc. 9th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jul. 2016, pp. 3–14.

[39] D. R. Reising, M. A. Temple, and M. J. Mendenhall, "Improving intra-cellular security using air monitoring with RF fingerprints," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2010, pp. 1–6.

[40] 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.

[41] M. D. Williams, M. A. Temple, and D. R. Reising, "Augmenting bit-level network security using physical layer RF-DNA fingerprinting," in *Proc. IEEE Global Telecommun. Conf. GLOBECOM*, Dec. 2010, pp. 1–6.

[42] T. Suzuki and N. Kubo, "GNSS-SDRLIB: An open-source and real-time GNSS software defined radio library," in *Proc. 27th Intl. Tech. Meeting Satell. Division The Inst. Navigat.*, Tampa, FL, USA, Sep. 2014, pp. 1364–1375.

[43] 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.

[44] C. Ellison, *SPKI Certificate Theory*, document RFC 2693, IETF Network Working Group, Sep. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2693.txt.

[45] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals," *Proc. IEEE*, vol. 80, no. 4, pp. 520–538, Apr. 1992.

[46] H. Dang, "Designed an MKR encapsulated in PDMS with sensitivity of -8.48 nm C," Tech. Rep., 2020.
[47] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford Univ. Press, 2004, pp. 295–319.
[48] D. F. Specht, "Probabilistic neural networks," *Neural Netw.*, vol. 3, no. 1, pp. 109–118, Jan. 1990.
[49] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold DSS signatures," in *Advances in Cryptology-CRYPTO '96* (Lecture Notes in Computer Science), vol. 1070. Cham, Switzerland: Springer, 1996, pp. 354–371.
[50] R. Berta, F. Bellotti, A. De Gloria, and L. Lazzaroni, "Assessing versatility of a generic end-to-end platform for IoT ecosystem applications," *Sensors*, vol. 22, no. 3, p. 713, Jan. 2022, doi: 10.3390/s22030713.
[51] D. Boneh and M. Franklin, "Efficient generation of shared RSA keys," in *Advances in Cryptology- CRYPTO '97* (Lecture Notes in Computer Science), vol. 1294. Cham, Switzerland: Springer, 1997, pp. 424–439.
[52] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2017, pp. 464–467.

JEAN G. V. ETIBOU (Student Member, IEEE) received the master's degree in computer applications from Cochin University of Science And Technology, Ernakulam, India, in 2015. He is currently pursuing the Ph.D. degree in computer engineering with the Mobile Computing and Networking Research Laboratory (LARIM), Polytechnique Montréal, Canada. He has been a member with LARIM, Polytechnique Montréal, since May 2019. His research interests include wireless networks communication and mobile payment in the IoT networks.

SAMUEL PIERRE (Senior Member, IEEE) is currently a Professor with the Department of Computer and Software Engineering, Polytechnique Montréal, and the Director of the Mobile Computing and Networking Research Laboratory (LARIM). He has authored or co-authored more than 600 technical publications, including articles in refereed archival journals, textbooks, patents, and book chapters. His research interests include wired and wireless communications, mobile computing and networking, cloud computing, and e-learning. He was a fellow of the Engineering Institute of Canada in 2003 and Canadian Academy of Engineering in 2008. In December 2011, he was appointed as a member of the Order of Canada. He has received several awards, including the Prix Poly 1873 for excellence in teaching and training (2001 and 2005), and the Knight of the National Order of Quebec in 2009. In May 2014, he received the Honorary Doctorate from the University of Quebec at Trois-Rivieres (UQTR) and the University of Quebec in Outaouais (UQO) in November 2016. In 2017, he obtained the El Fassi Prize from the Agence universitaire de la Francophonie (AUF) to highlight the action of a person who has exerted a significant influence through the quality of his expertise and the innovative nature of his achievements at the international level in the fields of research, training, development and international cooperation, governance and/or transfer of knowledge or skills. In 2020, he received the Grand Prize for Professional Excellence from the Order of Engineers of Quebec (OIQ). In 2021, he received the Gold Medal from Engineers Canada.