SEAT: Secure Energy-efficient Automated Public Transport Ticketing System

Chayan Sarkar, *Member, IEEE,* Jan Jaap Treurniet, Sujay Narayana, *Graduate Student Member, IEEE,* R. Venkatesha Prasad, *Senior Member, IEEE,* and Willem de Boer

***Abstract*—In smartcard-based travel payment systems, passengers have to place the smartcard near the journey registration devices once each for check-in and check-out to authenticate their travel. This is an annoying process when if the journey involves multiple stops. In this work, we describe a working system of *s*ecure *e*nergy-efficient *a*utomatic *t*icketing (SEAT) for public transport, which transforms traditional Check-in/Check-out system into Be-in/Be-out system. In SEAT, a Bluetooth Low Energy (BLE) enabled smartphone communicates with registration devices to track the journey for pricing without any human intervention. SEAT is vigilant towards the energy consumption of the BLE device under various conditions, and security and privacy threats to the overall ecosystem. We develop models for energy consumption and latency for BLE devices under the influence of mutual interference based on experiments with 32 BLE devices. We utilize these models to develop an energy-efficient protocol for SEAT that is secure and privacy preserving. Our experiments show that a BLE module consumes only 18.3 J daily under the proposed system model, which is less than 0.1% of the total capacity of a typical smartphone battery.**

***Index Terms*—Be-in/Be-out, automated public transport ticketing, BLE mutual interference, BLE energy consumption.**

I. Introduction

The Internet of Things (IoT) advocates for an ecosystem of connected devices anywhere and anytime [1]. This has kindled the integration of billions of embedded devices into our surroundings. Through smart applications, these connected devices promise to improve our lives by communicating and exchanging data seamlessly without any (or minimal) human intervention. An automated public transport ticketing system is one such application. In the sequel, we describe motivations and our contributions in the context of automated public transport ticketing system.

A. Motivation

Smartcard-based public transport ticketing systems are popular in the last two decades as it relives the commuters and the service providers from the burden of buying/selling tickets for each journey. However, the passengers have to register their journeys through check-in and check-out by placing the card near the tracking devices installed in the vehicles, at the stations, etc. The process becomes annoying for daily commuters if they have to use multiple modes of transportation, such as bus, train, tram, etc., or if the journey involves different segments. Many investigations have been done to improve the public transport ticketing system [2], [3], [4] using smartcards but none of these solutions is feasible for a fully automated ticketing system without any human intervention.

We propose a smartphone-based application that communicates with tracking devices and registers the journey automatically. Based on this, the cost of a journey can easily be charged to users. To detect whether a journey has started, ended or still continuing, a smartphone needs to communicate continuously with the tracking devices. Modern smartphones are equipped with various modes of wireless connectivity, e.g., WiFi, GSM, Bluetooth, NFC, etc. For this application, we need a wireless communication technology that offers omnidirectional transmission with a range of at least a couple of meters, low data rate, and low energy consumption. We selected Bluetooth Low Energy (BLE) is a technology for wireless personal area networks, which is designed for low energy consumption applications. Moreover, users are familiar with Bluetooth and using it more often. Thus, Bluetooth can be an ideal vehicle for this application.

B. Contributions and Outline

This article describes a working system of an automated public transport ticketing system, called SEAT (secure energy-efficient automatic ticketing). In this system, a passenger's smartphone automatically registers a journey by communicating with the tracking devices installed in the vehicles through BLE. In rest of article, the smartphone of a passenger and journey tracking devices are collectively referred as SEAT devices.

- In Section III, we describe the system architecture, which is different from traditional check-in/check-out based system and enlist research challenges that need to be addressed.
- In Section IV, we describe a communication protocol to register and track the journey of a passenger keeping an eye on security and privacy of message exchange between two SEAT devices; and also overall threat to the system. Our protocol has a novel authentication procedure that implements a combination of weak anonymity and unlinkability apart from perfect forward secrecy.
- Energy consumption by various applications is a major issue for smartphone users thus SEAT will be less appealing to the users if the energy consumption is high. In Section V, we investigate the impact of interference from close-by BLE devices on the energy consumption and latency by conducting experiments with 32 BLE devices. Then, we develop an energy consumption and latency

model. To the best of our knowledge, we are the first to conduct research regarding BLE energy consumption under mutual interference. Application developers can use our energy models.
- We evaluate our system in Section VI. Specifically, we ensure that the communication protocol for SEAT not only protects the passengers and the system from possible threats, but it also consumes low amount of energy from smartphones. As a result, the protocol can also be useful in other energy-constrained applications.

In summary, this article provides the complete details of the design choices and working of SEAT along with the extension of our previous study [5] on BLE energy consumption modelling.

II. Related Work

In 1993, McDaniel *et al.* [6] studied an automated system for fare payment in public transportation. Based on the available RF smartcard technology at that time, they found that such a system is too expensive to implement but they expected future technology improvements to make it feasible. Caulfield *et al.* [7] studied the requirements of passengers for vending machine-based public transportation ticketing system. The study revealed that one of the important wishes of the passengers is to be informed about cost, routes and estimated arrival times. Many research efforts have been done to improve the public transport ticketing system based on samrtcards [2], [3], [4] based on either RFID or NFC, which requires explicit check-in/check-out by the passenger. Thus, none of these solutions is suitable in a fully automated ticketing system where no human intervention is needed.

Many practical studies have also been conducted in various parts of the world about advanced public transport ticketing systems. A few examples of such studies are EasyRide by the Swiss Federal Railways Association [8], ComfoAccess from Trapeze in Leipzig, Germany [9], etc. These systems use active RFID requiring the explicit involvement of the passengers to register and de-register their journeys. However, our goal is to move away from these types of check-in/check-out system to more implicit system, where the system can automatically and implicitly track and register the length of the journey. We term such system as be-in/be-out (BIBO).

Nowadays almost everyone carries a Smartphone, which is a powerful computing device. Thus, it can be utilized for hassle-free implicit ticketing of the passengers. However, rampant energy drain from a smartphone is a great cause of concern. With the emergence of BLE, low-power data transfer within a proximity using a smartphone became possible making it the best alternative. The British Department of Transport has also considered BLE as a future technology for automatic ticketing system [10]. Being a powerful computing device, advanced security mechanisms can also be used easily in a smartphone based automatic ticketing system, which is essential for such a system.

One of the early efforts of a BLE-based public transport ticketing system was by Kostakos *et. al* [11]. The system was limited to collect only the source and destination matrix of each passenger journeys. Recently, Kuchimanchi [12] has proposed a BLE-based ticketing system that adds a custom profile on top of the BLE specification to accomplish the payment procedure not taking into account automated system and energy consumption. Moreover, limited attention is given to the security and privacy guarantee, which is a major requirement of such a system.

Similarly, Narzt *et. al* has conducted a feasibility study of BLE-based automatic public transport ticketing system [13] to check whether an implicit interaction between a vehicle and a passenger's smartphone using BLE communication. In the successive works [14], the authors developed a system for automatic tracking of the passengers for automatic ticketing. However, their system does not focus on the security and privacy aspects as well as an energy-efficient communication framework, however, we have developed a fully working prototype of the system. Along with the technical challenges of BLE-based communication, the energy consumption and overall security and privacy issues of the system are major factors. Our system extensively tackled these issues.

III. System Design

A schematic overview of the system can be found in Fig. 1. We envisage the complete system consisting of a cloud or database, where the details of all the signed up passengers are present. There could be multiple backends for the sake of scalability of the system. The central database keeps the information of all vehicles, ticket checkers, passengers, and keeps track of the performed journeys.

Before we explain the working of the system, we list here the broad requirements: (a) the system should be scalable; (b) the customer must be able to connect and authenticate securely; (c) the amount of energy used should be minimized; (d) the system should not leave any passenger unaccounted for his/her journey; and (e) further, the periodicity of checking passenger's journey should be balanced so as to not to drain the battery while not leaving a journey out.

The vehicle devices are connected to the backend server and many such servers may be required to address the scalability. Thus the systems can also be protected against compromise. The ticketing device in vehicles helps passengers checks-in or -out automatically. In Fig. 1, there are two passengers (Alice and Bob), and a ticket checker (Walter) apart from a malicious eavesdropper (Eve). We have used the terms ticket checker and guard interchangeably.

The system works as follows: Before a journey, the passengers need to have an application installed on their smartphone and register themselves with the backend. They also need to communicate from time-to-time (sparsely) with the backend to receive authentication information. While travelling, the device in the vehicle identifies a passenger and makes a local check-in for the authenticated passenger. Furthermore, it may supply information about the schedule and possible changes in the schedule to the passenger device. The ticket checker's device connects to the vehicle device to query it about the check-in status of passengers and connects to the passengers' smartphones to identify them. To register journey details, the

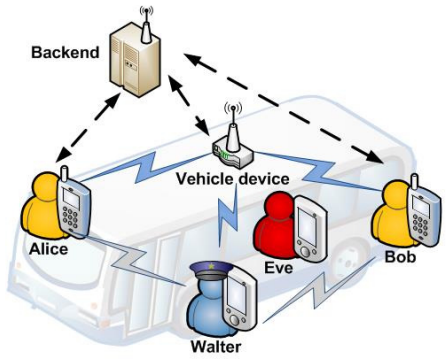


Fig. 1: Proposed system overview.

vehicle device can either connect to the backend throughout the day (for example via a 3G mobile data connection) or at the end of the day. Please note that an attacker can eavesdrop communication or perform any transmission to try to break the system or gain information about the passengers. Thus, the systems need to be secured using encryption and other security mechanisms.

A. Challenges

Security and privacy: Wireless communication used among the passengers, vehicle and ticket checker can be prone to eavesdropping. This can lead to losing sensitive and private data, monetary loss, etc. Thus measures have to be taken to ensure security and privacy and prevent abuse of the system. Attacks on the vehicle and guard device also need to be prevented and/or detected.

Energy consumption: Smartphone batteries have limited capacity. While wireless data transmission tends to be energy heavy and it intensifies with a number of concurrent passengers. Calculations, simulations, and experiments are needed to determine how much energy is consumed by the proposed system. A careful communication protocol design can reduce battery depletion of the smartphones of passengers.

Scalability and accuracy: Since the system takes care of journey payments, the accuracy of the check-in process is important. Billing of journeys that people did not make should be avoided and similarly, people should be rightfully billed for their journeys. The system will potentially be used by a lot of passengers at the same time; thus accurate billing with an increase in number of concurrent passengers is important as well.

Verification: A guard should be able to verify if a passenger claiming to have paid is indeed true. This means the guard's device needs to communicate with the passenger's device to exchange data.

Localization: To provide accurate billing information, the system needs to be able to make the distinction between a passenger who travels in the vehicle and a person who is within the reach of the vehicle device but not in the vehicle. Persons can be, for example, waiting at a bus stop for another bus or driving next to the bus in a car.

Apart from these challenges, others such as ease of use, the ON/OFF state of the mobile devices, etc., are to be considered. In this work, we focused only on the energy consumption by BLE devices and developed an energy-efficient communication protocol for SEAT, which is secure and privacy preserving.

IV. Communication Protocol

We start with a generic description of the protocol, whose main aim is to track the journey of a passenger. Then, we analyze the requirements regarding security and privacy aspects and propose a secure protocol. Finally, we provide implementation details of the protocol in a BLE device.

A. Global design

An overview of the protocol is shown in Fig. 2. The SEAT devices involved in the entire journey tracking process are described in the sequel below.

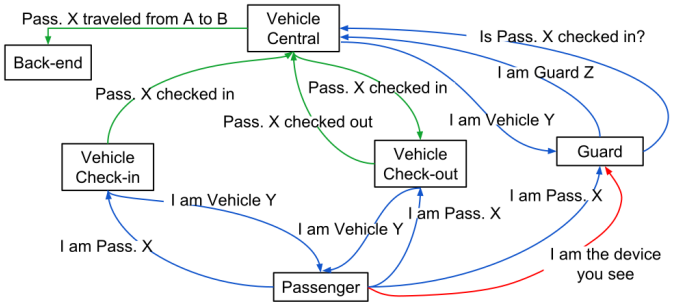


Fig. 2: Global design scheme of the proposed protocol.

Central backend: In a central backend, information about all journeys using all the vehicles is collected. Individual passengers are billed based on their journeys.

Vehicle check-in: The vehicle device is configured to check-in any device within the range that is considered to be inside. When a successful check-in is performed, the passenger is notified by a visual and audible signal. To make sure a check-in can be registered fast enough, an additional device can be placed near the bus/tram stops or at the railway platforms. This device notifies any passenger device within range to change its advertising settings for a certain period.

Vehicle check-out: The check-out device regularly checks if a passenger who checked-in is still present. If the passenger's device is not in the range for a certain period, the passenger will be checked-out.

Vehicle central: A single device can act as both check-in and check-out device. However, in a larger vehicle, there can be multiple such devices that cooperate to determine whether a passenger moved out of the vehicle or changed location within the vehicle. As a result, a centralized device (vehicle central) keeps track of all passengers present in a vehicle. It also communicates with the central backend to register journeys made by each passenger (online or offline).

Passenger's device: The passenger's device –a smartphone– accepts connections from the check-in/out devices and the guard's devices (to verify check-in).

Guard's device: The guard device is connected to the vehicle central device and only connects to passenger's device in a short range (chosen based on signal strength). The guard device checks with the vehicle central device if the passenger is indeed checked-in. The guard can verify manually (for example by a random code displayed on both the devices) if the connected device is the device shown by the passenger. To save energy for the passenger's device, the wireless connectivity could be turned off by default and turned on only when the accelerometer data suggests that the user is travelling by a public transportation vehicle. Several researchers proposed methods to detect transportation mode based on accelerometer data [15].

B. Security and privacy aspects

The proposed system must be secure against abuse while preserving the privacy of a passenger. We now define the requirements with regard to security and privacy.

Offline operation: The passenger's and guard's devices might not have a continuous connection with the central backend. This means a passenger will need to be authenticated using just authentication information that is available locally.

Storing long-term keys: To identify every unique passenger and ensure security and privacy within the system a number of secret keys are distributed among the SEAT devices. If the tracking devices – guard's or passenger's – are stolen, the secret keys stored in these devices may be extracted and misused. To restrict the impact to a certain period, the system should not allow any long-term secret authentication information to be stored in these devices.

Mutual authentication: The protocol must achieve mutual authentication between devices: a passenger needs to be sure that he/she is dealing with a registered vehicle or guard device and vice versa.

Anonymity and unlinkability: The passenger's identity must be protected except the tracking and guard's devices. Additionally, we also require unlinkability, i.e., an adversary should not be able to link two separate journeys performed by a passenger.

Perfect forward secrecy (PFS): When a vehicle or guard's device is disposed of, an attacker could extract the secret authentication information from the device. If this happens, we require that the attacker is not able to decrypt any previously recorded communications sessions. This requirement is known as *perfect forward secrecy* [22].

C. Protocol design for SEAT

Some of the existing literature regarding protocols that fit the requirements are summarized in Table I. Apart from these, the IETF specifies Transport Layer Security (TLS) 1.2 in RFC 5246 [23], which is widely used on the Internet. However, it is not optimized for the environments with limited resources. Additionally, PFS can be achieved in TLS, but anonymity and unlinkable authentication are not implemented. Though Song *et al.* provides a valuable guidance towards the security and privacy of cyber-physical systems [24], they do not provide a protocol suite that fits our need. Similarly, Zhou *et al.* outlines

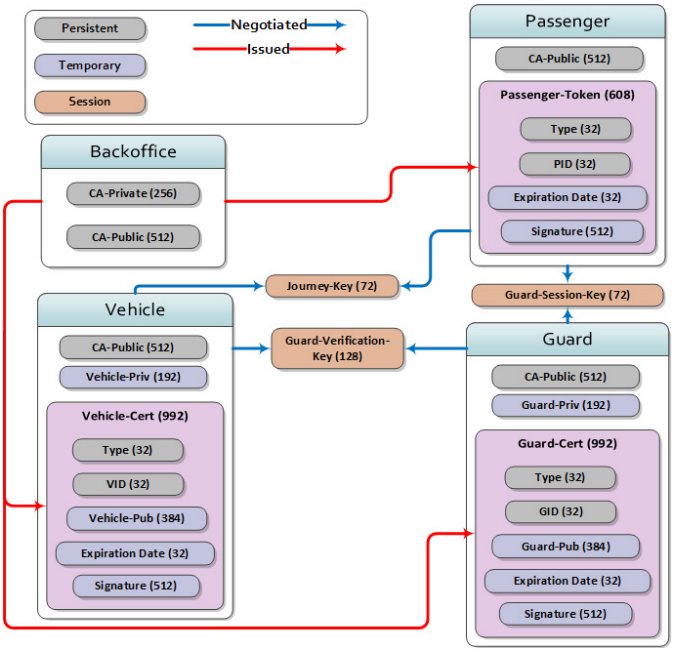


Fig. 3: Key distribution scheme along with the key size.

a number of security and privacy challenges for IoT [25]. To the best of our knowledge, none of the existing protocols implements expiration or revocation for the foreign server's key, and the required combination of weak anonymity, full unlinkability between sessions and PFS, which are the strong requirements of our system. SEAT uses a set of security keys among the devices, which play a vital role in maintaining secrecy and privacy. Next, we describe the key distribution in the system.

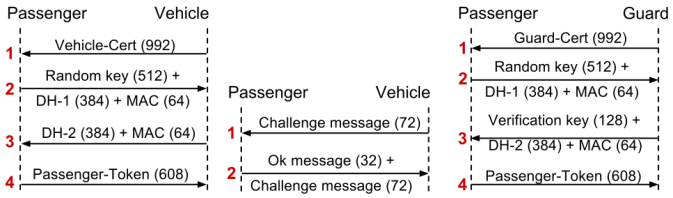
Key distribution: Fig. 3 shows an overview of the keys in the system and how they are distributed. The *backend* functions as a Certification Authority (CA); thus it has a persistent CA public/private key pair. This private key is used to sign certificates and tokens that are issued to the vehicles, guards and passengers. On the other hand, the public key is delivered to any registered device to verify signatures for tokens and certificates. Additionally, the backend keeps track of the revoked certificates and tokens.

The *vehicle central* (and the *guard*) device regularly generates its own public/private key pair. The public key is sent to the backend, to receive a public key certificate, which is valid for a limited period. It receives revocation lists of outdated/blocked guard certificates and passenger tokens. As a booting process, the *vehicle central* (and the *guard*) devices are issued by a central authority, who provides the initial pair of keys and certificates.

The *passenger* device periodically receives a token, which is valid for a short span and receives revocation lists for vehicle and guard certificates. Note that we use a public key certificate for the guard, but only a (simpler) token for the passenger. As the token has a smaller size, it requires fewer data to be transmitted by the passenger device. Though the vehicle and guard's devices will be connected to the backend once every day, a passenger might not have a data connection that often.

TABLE I: Overview of properties for existing secure authentication protocols for resource constraint devices.

Method	Authentication	Anonymity	Unlinkability	PFS	Key recency	Key revocation
Yang *et al.* [16]	Mutual	Weak	No	Yes	No	No
He *et al.* [17]	Mutual	Strong	Yes	Yes	No	Yes
Wang *et al.* [18]	User	No	No	Yes	No	No
Almuhaideb *et al.* [19]	Mutual	Weak	No	No	Yes	No
Li *et al.* [20]	Mutual	Weak	Yes	Yes	No	No
Liu *et al.* [21]	Mutual	Strong	Yes	Yes	Yes	Yes
Our protocol	Mutual	Weak	Yes	Yes	Yes	Yes



(a) check-in (b) presence check (c) verification

Fig. 4: Message exchange between a passenger and a vehicle during check-in and checking the presence.

Thus, we suggest issuing tokens with a validity of two weeks.

Authentication procedures: We describe the authentication and key exchange procedures for the different connections in the system.

Passenger - Vehicle: Fig. 4a shows the key establishment and authentication procedure between the passenger and the vehicle. First, the vehicle sends a certificate with his public key (*CertV*) to the passenger. Then, the passenger uses this key to encrypt a randomly chosen key (*R*) and sends this to the vehicle, together with the first part of the Diffie-Hellman (DH) key (*DH-1*) and a Message Authentication Code (MAC) created with key *R*. Then, the vehicle responds with the second part of the DH key (*DH-2*) and a MAC for this key is created with *R*. Now, both the parties can construct the journey key (*KeyJ*). Finally, the passenger uses this key to encrypt his secret token and sends this to the vehicle.

In Fig. 4b, the presence check procedure between the vehicle and the passenger can be found. The vehicle sends a random message, the challenge (*Chal*), to the passenger, encrypted with the journey key *KeyJ*. To prove he knows the journey key, the passenger device decrypts the challenge, adds an OK message, encrypts the results and sends this back to the vehicle.

Passenger - Guard and Vehicle - Guard: The key establishment and authentication procedure between the passenger and the guard is very similar to the one between the passenger and the vehicle, except for the verification key. The requirements are different for the authentication procedure between the guard and the vehicle. Though they need to establish a session key and perform mutual authentication, this communication does not require anonymity and unlinkability. Here, we omit detailed procedures for the paucity of the space.

Algorithm choices: In this section, we select the algorithms to be used for the implementation of the protocol.

Hash functions: For applying digital signatures, a hash function is needed. We select the SHA-256 hash algorithm as defined by [26] and recommended by [27].

Symmetric encryption: Where symmetric encryption is needed, we select the Advanced Encryption Standard (AES) [28] in CCM mode [29]. Both are recommended by [27] and support is included in the Bluetooth Core Specification. AES has a minimum key length of 128 bits. When the key exchange results in a shorter key, the selected hash algorithm will be used to inflate the key to 128 bits.

Asymmetric encryption: The most well-known asymmetric encryption protocol is RSA [30]. A huge disadvantage of RSA is the key size. For the required security level of 128 bits, a key length of over 3072 bits is required. Relatively new class of asymmetric encryption algorithms are those based on Elliptic Curve Cryptography (ECC). These systems are approved by the NIST for US government encryption [31]. Systems based on elliptic curves have the advantage of relatively small key sizes: twice the size of the asymmetric equivalent key length is advised by [27]. For this reason, we select ECC for our system. When using ECC, the following relations hold for the sizes of keys: the private key has the same length of the key size, and the public key is twice the keysize [32]. A plaintext block is twice the key size, and the resulting ciphertext is twice the plaintext size.

Message authentication codes (MAC): For message authentication codes in the protocol, we choose CMAC as described by NIST [33] and recommended by [27]. This MAC is AES-based and has a minimum recommended tag size of 64 bits.

Digital signatures: In the certificates and tokens, digital signatures are used. We use the Elliptic Curve Digital Signature Algorithm (ECDSA) as described and approved by the NIST [34]. This algorithm results in a private key of twice the required security level, a public key twice the size of the private key and a generated signature of twice the size of the public key [35], [36].

Key exchange: For secure key exchange, we need a key exchange protocol. The classic approach is the Diffie-Hellman protocol [22], but this has the same disadvantage as RSA: large key sizes and high computational complexity. The NIST describes Ephemeral Unified Model Elliptic Curve Cryptography Cofactor Diffie-Hellman [31]. This protocol uses ECC to reduce key size and complexity. For the key exchange in this protocol, both parties need to send an ephemeral public key, which is double the size of the key length [37].

Required security levels and key sizes: We determine the required minimum security levels for the keys used in the system based on the ECRYPT II yearly report on algorithms and key sizes [27]. This report gives advice on key lengths and encryption algorithms based on the required period of the

TABLE II: Overview of required security levels and resulting key sizes.

Key	Security	Key type	Minimum/used key size
CA	128 bits	ECC	256/256 bits
Vehicle	80 bits	ECC	160/192 bits
Guard	80 bits	ECC	160/192 bits
Session	72 bits	Symmetric	72/96 bits
Temp. auth.	64 bits	Symmetric	64/64 bits

TABLE III: For various protocol steps (s), message sizes in bits (b) and number of data packets (p).

(a) Check-in

s	b	p
1	992	7
2	960	6
3	448	3
4	608	4
total		20

(b) Presence check

s	b	p
1	72	1
2	104	1
total		2

(c) Guard verification

S	Bits	Packets
1	992	7
2	960	6
3	576	4
4	608	4
total		21

security and the available budget of possible attackers. The possible attackers range from 'Hacker' with a budget of $400, to 'Intelligence agency' with a budget of $400M.

For the keys where asymmetric encryption is used, we need to convert the symmetric equivalent security level to an asymmetric key length. When Elliptic Curve Cryptography is used, the recommendation from [27] is to use a key with twice the size of the symmetric equivalent. The CA key, used for signing certificates and traveller tokens, is constant and will not change over time. This means very long-term protection is needed. Thus we require at least 128 bits security. We select the NIST-suggested P-256 curve [34], which results in the private key size of 256 bits, public key size of 512 bits. The generated signatures are 512 bits long as well.

For the vehicle and guard keys, that are valid for a limited period, a short-term protection is sufficient. So, we require 80 bits of security, and thus, the minimum key length is 160 bits. We use the NIST-suggested P-192 curve here, which results in a private key size of 192 bits and a public key size of 384 bits.

The session key is renewed every session. Thus, breaking this key means information concerning only one journey is at risk. A short-term protection with a minimum security level of 72 bits is required for this. The temporary authentication key used in the authentication procedure is valid for only a very small time. So, this will only need to be safe against real-time attacks. We select a minimum level of 64 bits security for this key. The required security levels are summarized in Table II.
Remark: A link-level security along with the application layer security as explained above could be added. However, this will increase the message exchange between the devices, which affects the energy consumption. If any link level security needs to be added, the protocol will seamlessly work since our protocol is implemented at the application layer.

D. Protocol implementation

We now describe how the proposed protocol can be implemented for a BLE device. We start splitting the messages exchanged during various parts of the communication into BLE packets. We fix how often connections are to be made, role of BLE device and the way the BLE security modes are used.

Message sizes: Based on the required security levels and key sizes, we fix the actual size of the messages transmitted in the authentication, presence check and verification procedures. Fig. 3 shows the total size of various certificates based on their contents. Fig. 4 shows the contents of the messages at various steps while communicating and the resulting message sizes at each step is in the Table III.

BLE packet distribution: As BLE supports relatively small-sized packets, we need to split all messages in packets with a maximum size of 20 bytes. We calculate the number of messages for each step using $N = \left\lceil \frac{M}{160} \right\rceil$, where N is the number of packets and M the message size.

The required number of packets for every protocol step can be found in Table III. Note that in version 4.2 of the Bluetooth Core Specification, the maximum packet size has been increased. This means no packets have to be split up anymore making our protocol simple to implement and achieve higher throughput.

Continuous vs. ad-hoc connection: There are two options for using BLE connections in our system. The first option is to make a connection when a passenger is checked-in and keep this connection active until the passenger leaves the vehicle. The second option is to set-up a connection every time data needs to be exchanged and disconnect immediately after.

In the previous section, we found that 21 packets need to be transmitted for a verification action. To perform this action within the required time of 1 s, a short connection interval would be necessary. For this reason, we choose to establish a new connection when needed.

V. Energy Consumption by BLE Devices

As mentioned earlier, excessive energy consumption by a passenger device (smartphone) can be detrimental to SEAT. Thus, we conducted a detailed study of energy consumption by BLE devices under various circumstances. Table IV summarizes some of the existing work on energy consumption and interference for BLE and no previous work considered the influence of mutual interference from BLE devices on energy consumption and latency. We first simulated up to 100 BLE nodes, to identify discovery latency and associated energy consumption. Later, we experimented with 32 BLE devices to learn latency and energy consumption. A detailed study of energy consumption and latency is described in our previous work [5].

A. Experimental setup

The experiments were performed using BLE modules from Bluegiga [44], [45]. These inexpensive modules combine an 8051 microcontroller with a Bluetooth transceiver and are availbe as a USB development kit allowing for easy energy measurements. The modules can be programmed using BGscript, which is a module-specific scripting language. There are two 'devices under test': a central and a peripheral device,

TABLE IV: Overview of existing BLE-based energy consumption and latency experiments.

Method	Bluetooth type	Work description	Mutual interference	Validation
Liu *et al.* [38]	BLE	Discovery latency	No	Experiment
Stranne *et al.* [39]	Classic	Throughput modeling	Yes	Experiment
Kindt *et al.* [40]	BLE	All operations	No	–
Chong *et al.* [41]	Classic	Throughput & Energy consumption	No (from Zigbee)	–
Gomez *et al.* [42]	BLE	Throughput	No	Simulation
Siekkinen *et al.* [43]	BLE	Energy consumption	No	–
Our experiment	BLE	Energy consumption & various latencies	Yes	Experiment & simulation

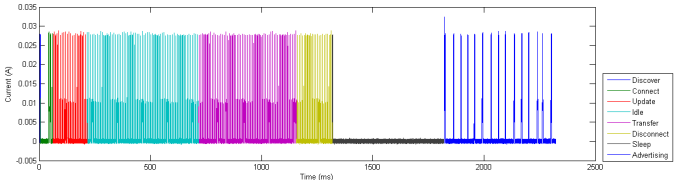


Fig. 5: Current consumption by the peripheral devices at the presence of 30 data transferring devices.

which are equivalent to a vehicle and passenger devices, respectively. Both are powered via a power supply and connected to an oscilloscope for current measurement. In addition, the central device is connected via USB to the Matlab application, and via a GPIO pin to a separate oscilloscope for sending trigger pulses. This helps in triggering guaranteed number of interfering devices.

Interference-generating devices: The interference-generating devices work autonomously in advertising mode and are only connected to a power supply. A control application is used to switch them to data transfer mode when needed. In data transfer mode, the devices are transmitting at the maximum achievable throughput, which is around 100kbps [40], [46]. We used up to 30 (central and peripheral included) interference-generating devices.

Channel limitation: During the experiments, we only used channels 1-8 for data transfer instead of all 37 channels to increase the influence of interference for the measurements to induce a significant impact on latency and energy consumption without needing an excessive amount of interfering devices. The number of advertising channels is not limited though, i.e., all the three advertising channels are used.

Testing environment: We performed the measurements in a meeting room in a corner of the building that was free from any electromagnetic interference of 2.4GHz band.

B. Experiment results

In this section, we present some of the measurement results. The complete and detailed measurement results can be found in [47].

Interference measurements: For various configurations, we measured the influence of an increasing number of interfering devices on the latency and energy consumption. Fig. 5 shows the current consumption by a pair of devices when 30 other devices were transmitting data. Fig. 6a shows the latency and energy consumption measurements for device discovery under interference from advertising devices. We see 2.2X maximum energy consumption increase for the peripheral device when 0 to 30 interferers are used. When comparing the simulation results with experimental results, we see that the trend is similar with a constant difference.

Fig. 6b shows the latency and energy consumption measurements for the connection setup operation under interference from advertising devices. Similar to the discovery results, we see a maximum energy consumption to increase 2.2X times when interfering devices varies from 0 to 30 devices. Fig. 6c shows the measurement results for data transfer under the influence of interference. In this graph, the net transfer energy is the amount of energy spent during the transfer of a data packet, reduced by the energy spent to keep the connection idle for the same period. In other words, the amount of energy spent on the transfer of data packet, assuming that a connection would have been active anyway. The net transfer energy is calculated as follows,

$$E_{net} = E_{transfer} - L_{transfer}\frac{E_{idle}}{L_{idle}},$$

where E_{net} is the net transfer energy, $E_{transfer}$ the measured transfer energy, $L_{transfer}$ the transfer latency, E_{idle} the idle energy and L_{idle} the time for which the idle energy is measured. From these graphs, we see that the net energy consumption increases by only 1.3X for the range of 0 to 30 interfering devices.

To make a choice between keeping a connection alive continuously and connecting again for every data transfer, we experimented with longer advertising and connection intervals. These measurements were done without any interfering devices. Fig. 7a shows advertising, connect and transfer latency for different intervals, whereas energy consumption by a peripheral device is shown in Fig. 7b. From these measurements, we can conclude that longer advertising intervals mean a lower energy consumption but also a higher latency.

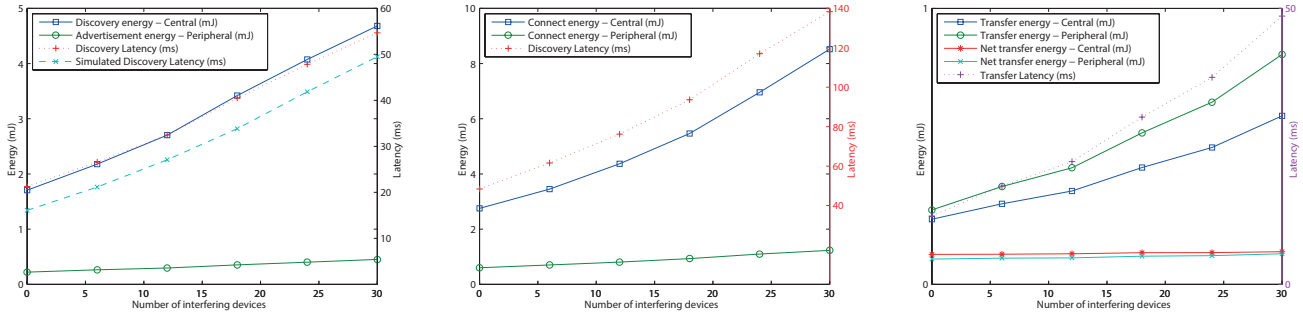
C. Model development

Based on the above results, we developed models for the energy consumption and latency of BLE devices. Here, we describe only the relevant models that are required for the proposed system.

Advertising: The advertising energy (in mJ/s) depends on the advertisement interval (I_a) and energy required to transmit one advertisement packet (E_a), and is given by the following equation,

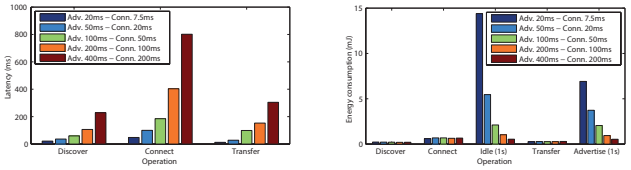
$$E_{adv}(I_a) = 1000\frac{E_a}{I_a + I_r}, \quad (1)$$

where I_r is the average random interval. From the experiments, we found $E_a = 0.22$mJ and $I_r = 10$ms.



(a) measurements for device discovery (b) measurements for device connect (c) measurements for data transfer

Fig. 6: Energy measurements for various operations of BLE communication.



(a) (b)

Fig. 7: Measurements for various advertising and connection intervals.

Discovery and connection *:* The discovery latency for a device depends on the advertisement interval (I_a) and the number of advertising devices (N_a). We assume the advertising interval is equal for all devices. The discovery delay (in ms) is given by the following equation,

$$T_{dis}(I_a, N_a) = (0.5I_a + T_{p,d})e^{\frac{2N_a}{3(0.5I_a+T_{p,d})}}, \quad (2)$$

where $T_{p,d}$ is the processing time for device discovery. From the simulation results, we found $T_{p,d} = 9.7$ms.

We model the connection latency (in ms) as the discovery time (T_{dis}) plus 3 times the connection interval (I_c). This connection latency includes the phase where successful connection is checked, and it is given by the equation,

$$T_{con}(I_a, I_c, N_a) = T_{dis}(I_a, N_a) + 3I_c. \quad (3)$$

Data transfer*:* We model the transfer latency for a packet with the maximum supported payload of 20B and is given by,

$$T_{tx}(I_c, P) = \frac{3}{2}I_c P, \quad (4)$$

where P is the number of packets. Similarly, we model the transfer energy (in mJ) for a packet as,

$$E_{tx}(P) = E_p P, \quad (5)$$

where E_p is the energy requirement for transmitting 1 data packet and its value is 0.27mJ (from the experiment).

VI. Evaluation

We evaluate SEAT from two different aspects - (i) security analysis of the developed protocol, and (ii) protocol evaluation in terms of energy consumption and a performance indicator.

A. Security analysis

There exist formal techniques to prove the security of protocols (e.g., a proposal by Fabrega *et. al* [48]. We do not apply those methods; rather, we give an intuitive explanation why the protocol meets the requirements and why it is secure against a number of known attack types. As there is a large similarity between the passenger-vehicle and passenger-guard procedures, we only do this for the passenger-vehicle procedure.

Mutual authentication*:* The vehicle reveals its identity through its public key certificate. Only a device knowing the secret private key corresponding to the certificate can decrypt the secret session key R. When the vehicle sends the MAC that it generated using R in Step 3, it proves to the passenger that it indeed is the vehicle that it claims to be. On the other hand, the passenger's token is secret and only known by the passenger. The vehicle validates the signature of the token. If the signature is valid, the passenger is authenticated.

Key establishment*:* As Diffie-Hellman key exchange mechanism is used to calculate the secret journey key (*KeyJ*), an eavesdropper knowing both *DH-1* and *DH-2* is still unable to calculate it.

Anonymity and unlinkability*:* The first time the passenger exposes information revealing his/her identity is in Step 4 by sending passenger token. However, this token is encrypted with the secret journey key. This means the anonymity of the passenger is protected. Consequently, the ciphertext changes with each session. This means an eavesdropper is unable to link two journeys made by the same passenger.

Man-in-the-middle attacks*:* In Step 1, an attacker is unable to modify the message because this would invalidate the signature of the certificate. In Step 2, an attacker could replace R by another value, but this would invalidate the MAC that is sent back in Step 3. An attacker cannot change DH-1 and in Step 3 because this would invalidate the MAC avoiding the induced change. In Step 4, an attacker is unable to make any changes, because he does not know the secret journey key *KeyJ*.

Replay attacks*:* The parameters for the Diffie-Hellman key exchange are generated randomly for every execution of the procedure by both parties. This means that any replay attack

will fail since the recorded messages are based on a different DH parameter.

Encryption implementation: For the authentication and key exchange procedure, it is not possible to use the BLE link layer security functions, because these are either insecure or need a symmetric key exchange in advance. From the moment the symmetric session key is established, there are two possibilities – either start using the BLE link layer encryption or keep using encryption in the application layer. By using the BLE link layer encryption, we make the implementation of the protocol easier since, according to the standard, this encryption is already supported by all the devices.

B. System performance requirement

In this section, we analyze the performance aspects of the system. These are quantitative properties of the system that ensure it is working sufficiently fast and scalable. We start by formulating the requirements and then test the developed models to determine if these requirements are met. The system performance requirements are summarized in Table V.

Description	Symbol	Maximum value	Unit
Check-in - Train	$t_{ci,a}$	30	s
Check-in - Bus	$t_{ci,m}$	500	ms
Check-out	t_{co}	30	s
Verification by Guard	t_v	1000	ms

TABLE V: Overview of quantitative requirements.

To determine the minimum time between two stations, we analyzed the timetable of HTM[1]. This public transport company in The Hague, Netherlands operates both buses and trams. We analyzed the schedule in terms of stations per minute. We found the tightest schedule for the following lines: bus 28 with 7 stops in 7 min, and tram 11 with 18 stops in 22 min. There are never two stops in the same minute, so we assume the minimum time between two stations to be 1 min.

A check-in needs to be processed at least before the vehicle reaches the next stop to make sure the right check-out location is registered. As the shortest time between two stations is 30 s, a regular check-in or check-out needs to be registered within this 30 s. When a passenger enters the bus, he needs to check in at the driver to allow the driver to check if every passenger checked in. To avoid delays when multiple passengers enter the bus, this type of check-in needs to be processed within 500 ms. When the guard checks a passenger, the devices must be ready for the visual verification within 1 s. We limit the number of passengers within the range of a vehicle device to 50. Mostly one device will be used in every compartment, but in the case of larger compartments, multiple devices may be used.

Remarks: In this work, we have only sketched the possible way considering the advertisement interval (and the energy spending therein) versus the accuracy in capturing all the journeys. The idea is to strike a balance between energy waste and fast check-in. Usually, there are additional devices installed at the stops/station. These devices can see any advertising

[1]We used the schedule as found on https://www.htm.nl/reisinformatie/ visited 2014-11-10.

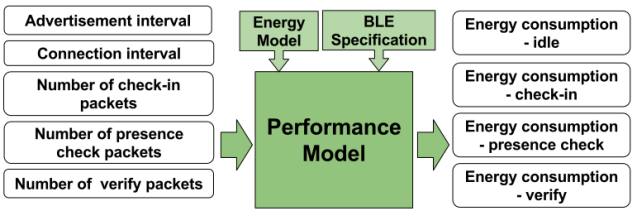


Fig. 8: Overview of the performance model.

passenger device. As soon as it establishes a connection after seeing the advertisement, it would indicate to the passenger that he/she is near the bus stop. This will indicate the passenger devise to increase advertisement frequency and immediately terminates the connection. No data transfer occurs with this connection set-up (indeed other business models could be added here, e.g., advertising some products, but this is out of the scope). After this quick connection set-up and closure, the passenger device can advertise at a higher frequency with a simple shared key based security mechanism to avoid unnecessary triggering. Even if a malicious device triggers the same, it would just increase energy consumption on the passenger device but sensitive data is not at risk.

C. Protocol evaluation

In this section, we present our performance model in terms of energy consumption. The energy model as presented in Section V-C is used as an input to this model. A schematic overview of the model can be found in Fig. 8. The model input, output, internal variables and symbols used in the equations are listed in the Table VI. In the model, a **check-in** is the initial

TABLE VI: Parameters used in the energy model and their values for a typical passenger.

Parameter	Description	Value
	Inputs	
I_a	Advertisement interval	1000 ms
T_{idl}	Idle time for a passenger device	23*3600 s
P	Number of packets	–
P_c	Number of packets - Check-in	20
P_p	Number of packets - Presence check	2
P_v	Number of packets - Verify	21
	Outputs	
E_{con}	Connect energy	0.6 mJ
$E_{adv}(I_a)$	Advertising energy consumption	0.22 mJ/s
$E_{tx}(P)$	Transfer energy (per packet)	0.27 mJ
$E_{idl}(T_{idl}, I_a)$	Energy consumption - Idle	–
$E_{chk}(P_c)$	Energy consumption - Check-in	6.0 mJ
$E_{pres}(P_p)$	Energy consumption - Presence check	1.14 mJ
$E_{vrfy}(P_v)$	Energy consumption - Verify	6.27 mJ

authentication of a passenger in a journey, where a temporary journey key is negotiated. A **presence check** is performed periodically to determine if a passenger is still present. A **verification** is an action performed by a ticket checker to check whether a passenger is successfully checked in.

When the passenger's device is idle, the energy is consumed only for transmitting advertisement messages,

$$E_{idl}(T_{idl}, I_a) = T_{idl} \cdot E_{adv}(I_a). \quad (6)$$

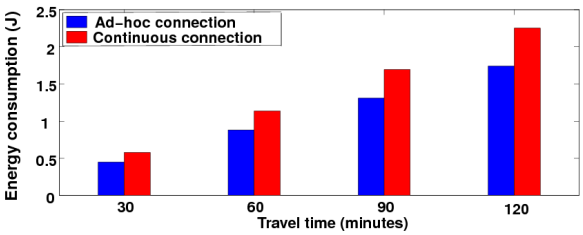


Fig. 9: Energy consumption comparison between continuous and ad-hoc connection for various travel duration. Energy consumption during the idle time is not considered here. A typical smartphone battery capacity is 26 kJ.

The energy required by the passenger's device to check-in is the energy required to connect plus the energy required to exchange the check-in packets,

$$E_{chk}(P_c) = E_{con} + E_{tx}(P_c). \tag{7}$$

The energy required by the passenger's device for a presence check is the energy required to connect plus the energy required to exchange the presence check packets,

$$E_{pres}(P_p) = E_{con} + E_{tx}(P_p). \tag{8}$$

The energy required by the passenger's device for a verification is the energy required to connect plus the energy required to exchange the verification packets,

$$E_{vrfy}(P_v) = E_{con} + E_{tx}(P_v). \tag{9}$$

Based on this performance model, we measure the energy consumption for the passenger device while using the system. To evaluate the resulting impact for passengers, we consider a 'typical passenger', i.e., (i) the passenger device remains idle for 23 hours, (ii) the person travels twice for 30 min in which his/her presence is checked every minute by the vehicle device, and (iii) the passenger is verified once by a ticket checker. The energy consumption (E_{daily}) for this passenger is calculated using the following equation,

$$\begin{aligned} E_{daily} &= E_{idl} + 2E_{chk} + 60E_{pres} + E_{vrfy}, \\ E_{idl} &= (3600 \cdot 23)E_{adv}. \end{aligned} \tag{10}$$

Applying the performance model for a typical passenger, we can conclude that the typical daily energy consumption of 18.3 J is negligible for an average smartphone battery (see Table VI for energy values). A smartphone with 2000 mAh and 3.7 V battery rating, has capacity of 26640 J. Thus, the application would consume about 0.069% of the total battery capacity of this phone and can be further reduced if the idle energy consumption is reduced. When a passenger is detected to be at home/office or the accelerator data suggests that the phone is not moving, the BLE advertisement process can be paused.

As mentioned in Section IV-D, we show that in Fig. 9 the energy consumption is higher for a continuous connection vis-á-vis ad hoc connections for various travel duration. Thus we choose to establish an ad-hoc connection each time a passenger device need to communicate

D. User Experience Evaluation

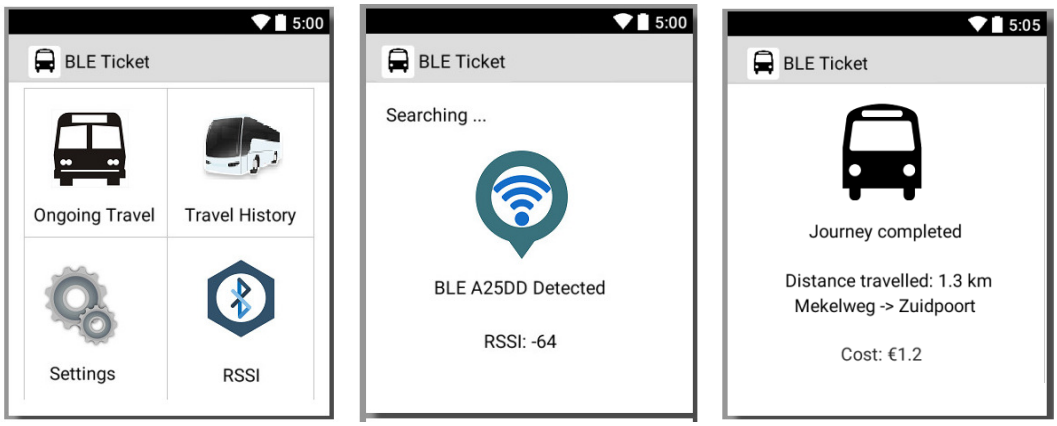
To test our system we developed a simple application with a server and a mobile *App*. We used a simple makeshift room of the size of a mini bus with a door and installed a BLE node. The screen captures from the App are shown in Fig. 10. Through the App, we provided an indication that the journey is recorded and when they get off, we show the total cost of the journey. Here we wanted to test purely the qualitative rating of our system and how users perceive our App. We asked 20 users to rate our system. Specifically, we asked passengers how they feel about this application by rating the App. Fig. 11a, More than 80% of the users scored the App idea to be 7 out of 10 and more than 50% scored 8 out of 10 (see Fig. 11a). Similarly, we asked the users to score accuracy in ticketing through this App and almost 70% of the users said they are highly satisfied scoring 8 out of 10 as shown in Fig. 11b. We definitely see the overall first impressions which are highly encouraging.

VII. Conclusions

We have proposed a Bluetooth Low Energy (BLE) based automated ticketing system (SEAT) for the public transportation system. We provided system requirements including a number of research problems. In this article, we focused our study on two major aspects – (i) energy consumption and interference aspect of BLE devices, and (ii) an energy-efficient protocol design for the payment system with emphasis on security and privacy aspects. We provided detailed design of the complete system.

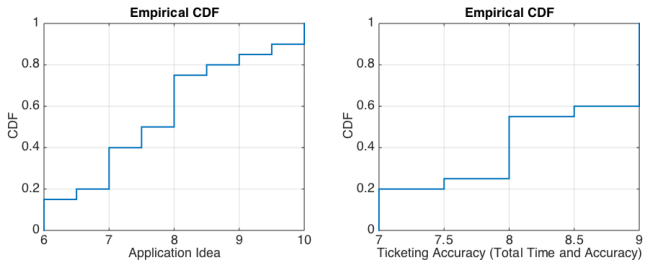
We performed experiments with 32 BLE devices for the energy consumption and latency under mutual interference. We developed models to predict the performance of the BLE devices, which is useful for application developers to predict energy consumption and study the influence of interference. Further, we used the model as an input for the development of the communication protocol. The protocol contains a novel authentication procedure that implements a combination of weak anonymity, unlinkability, and energy efficiency and can be useful in other applications as well. For a typical user who actively travels for an hour in a day, we find a total energy consumption of 18.3 J, which is about 0.069% of the capacity of a typical smartphone battery. With the latest version of the Bluetooth standard, this number can be decreased further.

SEAT can be used for automatic attendance collection of the employees in an organization. In this case, the energy consumption is not an issue since the attendance will be collected only once in a day (or twice if the leaving time is also registered). As a result, the BLE module can be activated only when the user is close to the proximity of the entrance of the premise. However, security of the system is utmost important for such system, which is also ensured by SEAT. Though we have tested SEAT with 32 BLE devices, the number is small to actually test the scalability aspects. Thus an immediate focus should be testing and improving (if required) the scalability of the system. An interesting extension would be to find the security break-point of the system and enhance the security features.



(a) simple app menu (b) vehicle identification (c) journey description

Fig. 10: The prototype implementation of SEAT and its working demo using a smartphone App.



(a) App idea perception (b) accuracy of the system

Fig. 11: The prototype implementation of SEAT and the user experience evaluation.

REFERENCES

[1] C. Sarkar, "Virtualizing the internet of things," Ph.D. dissertation, Delft University of Technology, nov 2016. [Online]. Available: http://www.es.ewi.tudelft.nl/phd-theses/2016-Sarkar.pdf

[2] P. T. Blythe, "Improving public transport ticketing through smart cards," *Proceedings of the ICE-Municipal Engineer*, vol. 157, no. 1, pp. 47–54, 2004.

[3] N. Mallat, M. Rossi, V. K. Tuunainen, and A. Öörni, "An empirical investigation of mobile ticketing service adoption in public transportation," *Personal and Ubiquitous Computing*, vol. 12, no. 1, pp. 57–65, 2008.

[4] R. Widmann, S. Grünberger, B. Stadlmann, and J. Langer, "System integration of nfc ticketing into an existing public transport infrastructure," in *Near Field Communication (NFC), 2012 4th International Workshop on*. IEEE, 2012, pp. 13–18.

[5] J. J. Treurniet, C. Sarkar, R. V. Prasad, and W. d. Boer, "Energy consumption and latency in ble devices under mutual interference: An experimental study," in *Future Internet of Things and Cloud (FiCloud), 2015 International Conference on*, Aug 2015.

[6] T. McDaniel and F. Haendler, "Advanced rf cards for fare collection," in *Telesystems Conference, 1993. 'Commercial Applications and Dual-Use Technology', Conference Proceedings., National*, Jun 1993, pp. 31–35.

[7] B. Caulfield and M. O'Mahony, "Passenger requirements of a public transport ticketing system," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, Sept 2005, pp. 119–124.

[8] T. Gyger and O. Desjeux, "Easyride: active transponders for a fare collection system," *Micro, IEEE*, vol. 21, no. 6, pp. 36–42, 2001.

[9] R. Zeller, "Trapeze renews lio operations control system in leipzig," Trapeze Computing Magazine, pp. 1–4, 2013.

[10] H. Lorenz, "Be-in-be-out payment systems for public transport," Final Report, GWT-TUD and Department of Transport, London, 2009.

[11] V. Kostakos, T. Camacho, and C. Mantero, "Wireless detection of end-to-end passenger trips on public transport buses," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 1795–1800.

[12] S. Kuchimanchi, "Bluetooth low energy based ticketing systems," Master's thesis, Aalto University, 2015.

[13] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselbock, and N. Hofler, "Be-in/be-out with bluetooth low energy: Implicit ticketing for public transportation systems," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 1551–1556.

[14] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselböck, and N. Höfler, "Bluetooth low energy as enabling technology for be-in/be-out systems," in *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*. IEEE, 2016, pp. 423–428.

[15] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, p. 13.

[16] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 1, pp. 168–174, January 2010.

[17] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 2, pp. 431–436, February 2011.

[18] Y. Wang, D. S. Wong, and L. Huang, "A one-pass key establishment protocol for anonymous wireless roaming with pfs," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–5.

[19] A. Almuhaideb, P. D. Le, and B. Srinivasan, "Two-party mobile authentication protocols for wireless roaming networks," in *Network Computing and Applications (NCA), 2011 10th IEEE International Symposium on*, Aug 2011, pp. 285–288.

[20] X. Li, Y. Zhang, X. Liu, J. Cao, and Q. Zhao, "A lightweight roaming authentication protocol for anonymous wireless communication," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 1029–1034.

[21] J. Liu, C.-K. Chu, S. Chow, X. Huang, M. Au, and J. Zhou, "Anonymous authentication for roaming networks with efficient revocation for large scale networks," *Information Forensics and Security, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[22] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1996.

[23] T. Dierks and E. Rescorla, "RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2," Tech. Rep., Aug. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5246

[24] H. Song, D. B. Rawat, S. Jeschke, and C. Brecher, *Cyber-physical systems: foundations, principles and applications*. Morgan Kaufmann, 2016.

[25] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based iot: challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, 2017.

[26] P. Gallagher, "Fips pub 180-4. secure hash standard (shs)," Gaithersburg, MD, United States, Tech. Rep., 2012.
[27] ECRYPT II Consortium, "ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012)," 2012. [Online]. Available: http://www.ecrypt.eu.org/documents/D.SPA.20.pdf
[28] National Institute of Standards and Technology, "FIPS 197: Announcing the Advanced Encryption Standard (AES)," Tech. Rep., 2001.
[29] M. Dworkin, "Sp 800-38c. recommendation for block cipter modes of operation: The ccm mode for authentication and confidentiality," Gaithersburg, MD, United States, Tech. Rep., 2004.
[30] J. Jonsson and B. Kaliski, "Public-key cryptography standards (pkcs) #1: Rsa cryptography specifications version 2.1," United States, Tech. Rep., 2003.
[31] E. B. Barker, D. Johnson, and M. E. Smid, "Sp 800-56a. recommendation for pair-wise key establishment schemes using discrete logarithm cryptography (revised)," Gaithersburg, MD, United States, Tech. Rep., 2007.
[32] J. van der Lubbe, *Basismethoden Cryptografie*. Delftse Universitaire Pers, 1994.
[33] M. J. Dworkin, "Sp 800-38b. recommendation for block cipher modes of operation: The cmac mode for authentication," Gaithersburg, MD, United States, Tech. Rep., 2005.
[34] C. F. Kerry, A. Secretary, and C. R. Director, "FIPS PUB 186-4 Federal Information Processing Standarts Publication Digital Signature Standard (DSS)," 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.362.5590
[35] N. S. Agency, "Suite b implementer's guide to fips 186-3 (ecdsa)," Tech. Rep., 2010.
[36] ISO/IEC, "14888-3:2006: Information technology - Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms (ISO/IEC," International Organization for Standardization, Geneva, CH, Standard, December 2006.
[37] N. S. Agency, "Suite b implementer's guide to nist sp800-56a," Tech. Rep., 2009.
[38] J. Liu, C. Chen, Y. Ma, and Y. Xu, "Energy analysis of device discovery for bluetooth low energy," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*. IEEE, 2013, pp. 1–5.
[39] A. Stranne, O. Edfors, and B.-A. Molin, "Experimental verification of an analytical interference model for bluetooth networks," in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*. IEEE, 2006, pp. 1–5.
[40] P. Kindt, D. Yunge, R. Diemer, and S. Chakraborty, "Precise energy modeling for the bluetooth low energy protocol," *arXiv preprint arXiv:1403.2919*, 2014.
[41] J. W. Chong, H. Y. Hwang, C. Y. Jung, and D. K. Sung, "Analysis of throughput and energy consumption in a zigbee network under the presence of bluetooth interference," in *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*. IEEE, 2007, pp. 4749–4753.
[42] C. Gomez, I. Demirkol, and J. Paradells, "Modeling the maximum throughput of bluetooth low energy in an error-prone link," *Communications Letters, IEEE*, vol. 15, no. 11, pp. 1187–1189, 2011.
[43] M. Siekkinen, M. Hiienkari, J. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, April 2012, pp. 232–237.
[44] Bluegiga, "BLE113 Datasheet," https://www.bluegiga.com/, 2014, accessed: 2014-10-14.
[45] ——, "BLE113 Development Kit 2.0 Datasheet," https://www.bluegiga.com/, 2013, accessed: 2014-10-10.
[46] ——, "Knowledgebase: Throughput with bluetooth smart technology," https://bluegiga.zendesk.com/entries/24646818-Throughput-with-Bluetooth-Smart-technology, 2013, accessed: 2014-07-21.
[47] J. J. Treurniet, "From check-in/check-out to be-in/be-out: Ble-based automated journey payment in public transportation," Master's thesis, Delft University of Technology, 2015.
[48] F. Thayer Fabrega, J. Herzog, and J. Guttman, "Strand spaces: why is a security protocol correct?" in *Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on*, May 1998, pp. 160–171.

Chayan Sarkar received the B.E. degree in computer science and engineering from Jadavpur University, Kolkata, India, in 2009, the M.Tech. degree in computer science and engineering from the Indian Institute of Technology Bombay, India, in 2011, and the Ph.D. degree from the Delft University of Technology, The Netherlands, in 2016. He is a Scientist with TCS Research, Kolkata. His research interests are multirobot collaboration, embedded systems, and Internet of Things.

Jan Jaap Treurniet received the B.Sc. degree in computer science and engineering and the M.Sc. degree in embedded systems from the Delft University of Technology, The Netherlands, in 2011 and 2015, respectively. He is a Consultant with Technolution BV, The Netherlands.

Sujay Narayana received the M.Sc. degree (Hons.) in embedded systems from the Delft University of Technology, The Netherlands, in 2015, where he is a Ph.D. Researcher. He was also an exchange student with ETH Zurich, Switzerland, during the masterâĂŹs studies under IDEA League Scholarship. His research interests are in the area of Internet of Things in Space (Space IoT), embedded real-time systems, and nano/femto satellites.

R. Venkatesha Prasad received the B.E. degree in electronics and communication engineering and the M.Tech. degree in industrial electronics from the University of Mysore, Mysore, India, in 1991 and 1994, respectively, and the Ph.D. degree from the Indian Institute of Science, Bengaluru, India, in 2003. He is an Assistant Professor with the Delft University of Technology, The Netherlands. He is a Distinguished Lecturer of IEEE and a Senior Member of ACM. He was selected by the External Affairs Minister of India as one of the 10 leading Indian scientists outside India to discuss on "Developing Cyber Capacity of India".

Willem de Boer received a Bachelor degree in electronics from the Noordelijke Hogeschool Leeuwarden, the Netherlands, in 1993 and a Master degree in computer science from the University of Twente, the Netherlands, in 1996. Since then he has been working at Technolution, a Dutch technology integrator, mainly in the field of electronic payments and ticketing. He currently is Principal Consultant and Security Architect at Technolution.