Secure Privacy Preserving Information Beacons for Public Transportation Systems

Thorsten Schulz, Frank Golatowski, Dirk Timmermann
Institute of Applied Microelectronics and CE, University of Rostock,
{Thorsten.Schulz,Frank.Golatowski,Dirk.Timmermann}@uni-rostock.de

***Abstract*—Bluetooth Low Energy Beacons currently provide a great potential for indoor localization to users with smart devices. This paper implements a security concept to mitigate weaknesses of current beacon concepts against forgery and request tracking. The method authenticates beacons with dynamic data by attaching secure signatures. The architecture is especially useful as a many-to-many solution. A test setup shows applicability in a public transportation vehicle to accompany the passenger information system and traveller route planning. A setup on a tram equipped with up to five beacons provided securely authenticated passenger guidance in and around the tram, referencing even a user's device in a pocket.**

Acknowledgment: I would like to thank the Rostocker Straßenbahn AG (Tram Service of Rostock) for the uncomplicated support of the test measurements.

I. Introduction

While navigation systems have seen mass adoption in cars in past years, mapping and navigation has also evolved to more general purpose mobile devices like phones and tablets due to improved computational power. With these devices, outdoor positions are acquired easily based on GPS signals and recently by identifying telephone and wireless networks.

In contrast, indoor localization sources its current growth from advances in inertial sensing and the deployment of Bluetooth Low Energy beacons. Many readily available indoor localization products using bluetooth LE technology are based on the "iBeacon" concept, sending out a beacon specific unique identifier for database look-up by the user application. This database is either static on the user device or an Internet based service. The latter leads to data access delays and inherent tracking by the service provider. Whereas a beacon sending out plain data sets could easily be forged.

Navigation in dynamic contexts pose additional requirements to beacon systems. Service status data could be sent instantly to the users device, reducing latency for Internet requests. These dynamic environments with anonymous user interaction are typically public facilities, e.g. universities, hospitals and especially public transportation vehicles. In this case navigation can be customized to specific user needs, i.e., users with reduced mobility (PRM). To prove the feasibility of the concept, after discussing the technical details in the next sections, Section V provides measurements taken on a real tram.

A. BLE Broadcast

Bluetooth Low Energy (BLE) device discovery has been drastically streamlined from classic Bluetooth. Devices with the peripheral role, in contrast to the central device which is in most scenarios the user's hand-held device, send out an advertisement to be discovered. This advertisement holds the id, name and whether it is connectable. This packet may also contain information about provided GATT-services on connection and other data fields. Peripheral devices can usually only connect to one central device and also stop advertising on connection. In Broadcast Mode the device is unconnectable and permanently sends out the advertisement packets filled with service related data to surrounding listening central devices. Due to the low complexity of the broadcast signal, it can be very energy-efficiently monitored.

II. Related Work

A. BLE Indoor Navigation

The findings on precise locations based on received signal strength indication (RSSI) from BLE beacons in the measurements related to this paper were not as positive as in [1], stating results of errors down to a meter. A related publication [2] shows more real-world results on one-shot fixes at an error within a few meters. The published presence of fast fading has been well noticed in this paper's measurements. As a result, further location estimates based on RSSI have not been incorporated. On top of this, we have also noticed strong fluctuations in RSSI based on orientation of the device and the user's hand position.

The authors of [3] introduce a smart-phone based indoor localization concept utilizing inertial navigation and Bluetooth beacons. They target especially complex stations of public transportation facilities. These buildings with their different level structures, stairs, lifts, escalators and elaborated distances tend to challenge travellers, especially foreign guests and travellers with limited abilities. This paper presents a viable extension to this concept with dynamic data being sent by the secured beacons from within a vehicle (tram).

B. Beacon concepts

One already quite famous implementation of BLE beacons is "iBeacon" by the company Apple. These beacons use a proprietary advertisement service packet to announce themselves with a UUID as their data payload. The user can set custom actions in the vicinity of an iBeacon but mostly custom apps will look-up specific information related to the beacon. The beacons are independent battery powered tags, usually without any configuration option.

[4] quotes estimates from 2014 that see 60 million beacons to be installed before 2020. These will be employed especially

in retail areas and for big data collection. Depending on local authorities, different laws govern the use of personal data. When used as an offline app, for example a digital guide in a museum, privacy is only at concern if malicious software is able to filter the location data mentions [4]. But when it comes to online apps, [4] cites German laws that require at least the users approval.

Even if consent is given, most users will not appreciate unreasonable data collection. Most server based solutions are not transparent to the user in terms of tracking. For better acceptance of applications based around public services this form of mistrust should be avoided.

Just recently another general-purpose concept ("Eddystone") for beacons has been published by Google as a patent (see [5]) and also as an open library with a license granting free usage rights. Google's publication addresses the current general problem of impersonation or other spoofing attacks to BLE beacons. The provided solution uses ephemeral identifiers and message integrity codes to authenticate the broadcasting device. The "validity of the message integrity code is gained by matching the message integrity code against a plurality of expected message integrity codes" previously generated. The verification of ephemeral identifiers may be separated from the observer which is the user's device.

The frame types for these "Eddystone" beacons are currently a UID, a URL and a status frame. The UID is composed of 10+6 bytes of namespace and instance-id. The URL can take a short URL, after allowing, e.g., 5 characters for the domain name, leaving 12 bytes for a resource identifier.

The message integrity code is computed by a 128bit-AES-block-cipher, transmitting only the lower two bytes of 16 over air. This coding scheme requires that the secret (rotating) key is also known by the verificator. This implies, from a security point of view, that the verifying module containing the secret key is not part of the user device, as user devices cannot be trusted. As a result web-based verification look-ups are required or, when limited to static messages, pre-computed message integrity codes must be pre-loaded to the user device.

C. Broadcast Security

Securing broadcast data means to authenticate and trust its source. Impersonation of broadcast beacons can be done with any BLE device supporting the peripheral role. Latest versions (2015) of Android and iOS support this broadcast functionality. Even though misleading data broadcast by malicious beacons may be of safety impact – if it can be done, it will be done and it will render the beacon service unusable, while being hard to trace the source of mischief.

For centuries signature methods have provided the basis for message authentication. Transferred to digital systems, *digital* signatures build upon asymmetric cryptography. As discussed in [6] and [7], due to high numerical complexity, signature algorithms are not well spread in the embedded or sensor network domain. There is no question whether to use a simpler cryptographic algorithm. But not just the algorithmic complexity is an impediment, also the length of the signature to be appended to the data payload is a handicap. Under these circumstances we have examined elliptic-curve signatures to sign beacon broadcast data.

Since elliptic-curve cryptography was published in 1985 (see [7]), it has matured in efficiency and proven in security applications. Still typical NIST-P256 implementations are merely feasible on battery powered devices. This paper presents an available hardware ECC-encryption device. Thanks to availability of a software-based implementation of the X25519 algorithms based on the Curve25519 in the Networking and Cryptography library (NaCl, pronounced "salt") by Daniel J. Bernstein et. al. in [6], this open, reviewed cryptographic library has been given preference. Further decisions are discussed later. In detail the astoundingly optimized version for microcontrollers by Hutter and Schwabe (see [8], [7]) was used on the prototype beacon.

III. Implementation of a "Salted" Beacon

This section describes the system of the "salted" beacon prototype. The "salted" attribute refers to the abbreviation of the Networking and Cryptography library (NaCl) to spice up the authenticity of a beacon with a secure signature. This added function requires the beacon prototype to provide a freely programmable microcontroller to generate the required cryptographic tag. The following section will also discuss alternatives to NaCl.

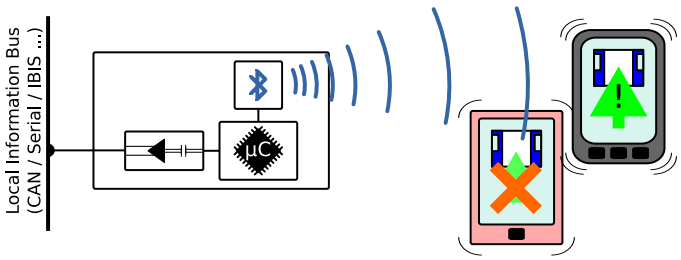

Figure 1. General hardware architecture. The beacon device is connected to a local data bus and broadcasts its data to listening devices. Depending on the user's chosen destination they signal further individual action.

The basic principle of a beacon with dynamic data broadcast requires the beacon to also connect with a local data bus such as RS-232, RS-485, CAN, Ethernet or alike. It could also have a simple I/O-interface to directly correlate certain packets with inputs from the connected appliance. But in the later shown test on a public transport vehicle (tram), the well standardized IBIS-bus was available. It is a core bus of passenger information systems (information displays, ticket machines) since the mid-80s.

The beacon readers are up-to-date smartphones, capable of BLE scanning. Signature verification is done using a pure Java software implementation of NaCl. Thus, the concept and technical requirements are not limited to a certain device brand. The proof-of-concept prototype software was implemented for an Android device only. The specific test devices will be discussed later in Section IV.

A. Prototyping Hardware

There are different types of BLE prototyping hardware available. One of the early well available hardware manufacturers was Nordic Semiconductors, among others. Most notable for quick prototyping are the Nordic Semi nRF51822 Smart

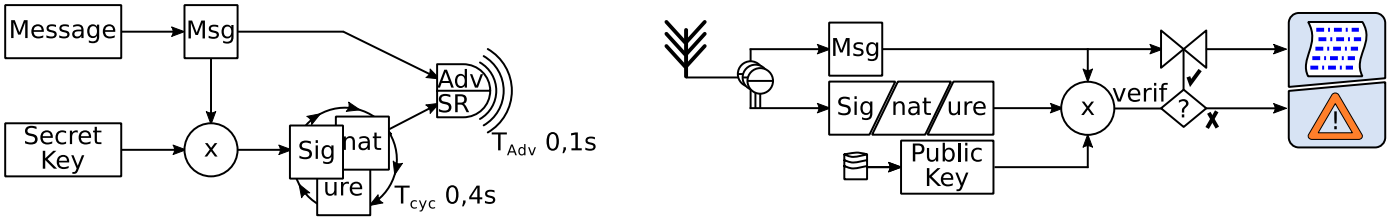

Figure 2. Signature data flow: Signing is done to the left on the beacon device, splitting up the signature across cycling scan response packets. The signature is reassembled and verified on the user device. (SR = Scan response)

Beacon Kit and different Arduino platforms. When the initial idea for the secure beacon concept sparked in early summer 2014, SoC development kits were sparsely available. So a previously engineered compact microcontroller module was fitted with a socket for the Olimex BLE module with radio module nRF8001 by Nordic Semiconductors.

To support the envisioned vehicular inputs for the test in Section V, a baseboard with wide range power supply and serial bus interfaces was added. The microcontroller itself is a general-purpose Atmel AVR XMega32E5 8-bit RISC device at 32 MHz. It was chosen due to its low complexity, low cost, previous application and adequate resources. The microcontroller's datasheet accounts for 7 mA current consumption in full active mode. Together with the BLE-radio module, voltage regulation and other components the average current consumption is slightly over 10 mA. While energy optimization was not a goal at this time, it gets clear that beacons with application-dynamic data have notably higher power consumption, than battery powered static beacons. From the point of use this is not an issue as the related appliance generally has some sort of land line connection.

B. Encryption

Very early trials with crypto implementations from "AVR-crypto-lib" (http://avrcryptolib.das-labor.org/) were not satisfying in terms of performance. The development of the library seems stalled. More promising was the integration of a hardware based crypto IC the Atmel ATECC108 (see [9]) providing Elliptic Curve asymmetric cryptography (ECC) and secure hardware key storage. The device supports Elliptic Curve Digital Signature Algorithm (ECDSA) using NIST Standard P256 curve and a high quality FIPS-random number generator. ECDSA is generally used in conjunction with SHA256 message hashing, which needed to be implemented in software. The recently revised version ATECC108A was extended with a hardware SHA256-algorithm.

Unfortunately, documentation of the ATECC108 device is under NDA and was not available. The sample libraries provided in exchange were a nuisance to debug on a different microcontroller target without the library documentation. The contained examples were not self-explanatory to get ECDSA signing running in an acceptable amount of time (well over a week). Further integration of the ATECC108A chip was abandoned temporarily in favour of the crypto-library μNaCl from [7].

Even though the reader is invited to become more acquainted with elliptic curve cryptography, it must be emphasized that NIST-P256 ECDSA and Curve25519-based Ed25519-algorithm – therefore also named *ED*DSA, are different and incompatible algorithms.

This work has not evaluated whether storing a secret key for EDDSA calculation within the AVR's EEPROM is secure. There are measures to lock off debug and programming interfaces, but their effectiveness has not been trialled. A proven advice on how often to generate new keys cannot be given at this point. Also, without any further means, secret keys must not be generated on an AVR, as it does not provide a qualified source for random numbers.

C. Over-The-Air Signature Transmission

BLE advertisement packets can carry up to 31 bytes of advertisement data and an additional 31 bytes in the Scan Response (SR) packet. The advertisement packet must carry the Flags field and the Local Name field. Even though the name can be shortened to the minimum of one byte, it is advisable to provide a generically traceable identification. This application uses 6 characters for the name. In this configuration there are 16 bytes left for the actual dynamic data in the advertisement packet (e.g. see table I below). The scan response is requested by the user device after receiving the the advertisement packet. The Nordic NRF8001 BLE module limits its service data fields to a length of 20 bytes, so the dynamic data in the scan response packet must be split up into two fields, in this case 3 bytes and 20 bytes plus each field's header(4 bytes).

The length of the chosen EDDSA signature is 64 bytes. These signature bytes could be segmented across two consecutive packet pairs and still leave 14 bytes for the actual data message. The disadvantage is, that message and signature data splice into each other. The favoured approach is to send the data message in every advertisement packet and break down the signature across three scan response packets. This gives the current message higher availability in noisy conditions, while the signature is pushed back to actual scan response requests. It must be noted though, that the user device cannot request a specific scan response packet. It is up to the secure beacon to frequently cycle through the three segments of the signature (see Figure 2). The test implementation uses an advertisement period of 100 ms (standard minimum) and cycles the signature segment every 400 ms. This can achieve a minimum reassembly time for a signed message in just under a second.

D. Beacon Application

An authenticable BLE broadcast beacon for dynamic data needs to read and convert the local appliance's (e.g. vehicle)

data, calculate the signature and update the data in the broadcast buffer. Unfortunately, in the actual executed on-vehicle tests, the proposed connection to the vehicle data bus was not established due to time constraints. So the message data packets contain static information about the beacon's in-vehicle location and an increasing 24-bit-time-stamp with a 2-second-resolution. This time-stamp could prevent plain replay attacks for a year before the secret key needs recreation.

The beacon software uses two threads: one background thread and one time-triggered thread. This dual approach is required to keep the system responsive throughout cryptographic calculations. The call to the NaC-library's function `crypto_sign_ed25519(sig_buffer, sig_len, msg, msg_len, secretkey)` is spawned into the background thread taking 0.95 s. It is interrupted by the higher prioritized timed-thread.

As a consequence, update of the data message is limited to minimum intervals of a few seconds. Depending on the urgency of the new message and the age of the previous message there are two ways to handle the calculation delay:

- broadcast new message data immediately and hold back the signature until it is updated, or
- hold back the new message data, and keep on sending the old data with its signature until the update is processed

In the tests, the first approach was taken, as the data update and signature calculation interval was fixed to 20 sec (to emulate dynamic data), which leaves the receiver with plenty signature packets before the next update.

IV. Message Reception

The receiver needs to activate BLE scanning and subscribe to notifications of received advertisements. The received packets need to be dissected for message data and signature segments. Once all three consecutive signature segments are received the verification can be carried out and the verified message can be presented to the user.

The packet evaluation and signature verification is part of the specific application. This application is also in charge of updating the public keys for verification via a secondary secure channel from a trusted key server. Compared to the earlier mentioned approach by "Eddystone" beacons, this would be completely *independent* of actual beacon sightings and could be in the frequency of once a month. Tracing key polling to beacon sightings is not inherent. The application would also have the chance to reduce message verification to random sampling, first sighting or non at all. Of course this breaks full security but could drastically save device energy in appropriate situations where this is justified (Figure 2).

A. Beacon Capture with Android

BLE support has seen great changes from early implementations in Android version 4.3, past 4.4 to version 5. Since the available devices did both run Android 4.4 at time of application implementation, the choice fell for API level 19. This API level does not provide a great deal of functionality for broadcast scanning other than providing the bare advertisement byte array in a discovery notification. This array already contains the data from the scan response packet (SR in Figure 2), so it needs not to be requested explicitly.

A noted disadvantage is, that some devices cannot deal with dynamic advertisement data. Once a beacon's advertisement is received, no notification about following packets is posted. As this was the case for one device in the test (Sony Xperia X1C), the application toggled the scan mode in short periods to acceptably circumvent this behaviour. The other device, a Samsung Tablet, did not show this issue and forwarded every advertisement packet.

The signature verification was done using a pure Java implementation of Ed25519 based on the ref10 implementation in SUPERCOP (see [10]). The library was forked from a reviewed repository: (https://github.com/str4d/ed25519-java). The verification run-time was not explicitly measured but was also not experienced to be of any significance.

V. Measurements, Results

An initial test was executed, to estimate how many beacons should be distributed in the latter test. The following test tries to show whether the concept is feasible to accomplish the following use cases, where the user is notified:

1) when vehicle reaches a transfer/exit station
2) if an approaching vehicle runs to ones chosen destination
3) guidance to find an on-board toilet or ticket machine
4) where to enter the vehicle for special areas

To cover these use cases, the structured message as described in table I was crafted to carry general and special items. As mentioned in preceding sections, the connection to the vehicle bus was not established since the tests were run in the maintenance shed not to interfere with actual service at this stage. As such, data other than the time-stamp was static. Though the five distributed beacons did vary in their on-vehicle position information and their relation to the ticket machine (use case 3) marked with "T" in Figure 4.

member	range	content
		vehicle data
sub-type	16	identify special sub type message
provider	4000	transport provider company id
route	65000	service relation data
course	250	service relation data
run	250	service relation data
station	$4 * 10^9$	id of upcoming / current station
delay	+/-120 min	current service timeliness
timestamp	1 year	current time of year (incr. in test)
		per beacon data
position	8	rel. beacon position in $\frac{1}{8}$th of carr.
carriage	16	nr. of carriage in train
reverse	flag	whether carr. is running reverse
door state	8 states	nearby door (n/a, open, locked)
find-ticket	4 states	n/a, go to front, to back, is here
find-WC	4 states	n/a, go to front, to back, is here

Table I. Example 16-byte message data for a beacon in a public transportion vehicle

A. *Tram Test Measurement*

The test was run in a static setting in the central tram maintenance shed. There were other tracks to the left and the right, but no other trams in parallel. The floor was made of metal grid panels and there were other metal constructions around the shed. The impact on wide range signals cannot be quantified, but disturbance was certainly present.

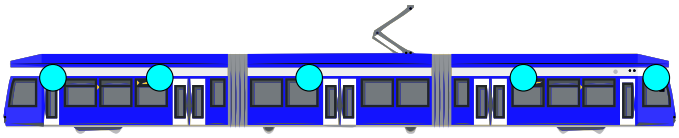

Figure 3. Side view of beacon positions that were affixed inside behind voute paneling.

The tram, that was available to be equipped with beacons for the measurement, was 30 m long, 2.3 m wide, had 5 doors on the right side and was an about 20 years old type "6NGTWDE". The beacon circuit boards were fixed in a plastic housing and taped to plastic or metal cable guides in the equipment cabinets between window and roof panelling. The cabinet doors, so called voutes, were made of fibre glass material and were closed for the tests. It was noticed in the results, that the placement of the beacon nodes must be selected carefully. For example, the third beacon was unknowingly placed very close to a loudspeaker which seemed to cause a negative impact. On the other hand, the fifth beacon was placed in the door mechanics compartment, which raised signal strength when the door was open. Since this effect was not planned for, the test logs were not annotated for the door state and doors were generally closed.

The measurements were captured with two user devices in parallel. The Samsung tablet was held tucked close in front of the body. The Sony mobile phone ran the same program with logging while kept in the right pocket of the pants. This is noticeable in a way that beacons to the left are seen less likely. Measurements outside were taken facing perpendicular to the track standing upright. Within the tram, the middle seat was taken, looking to the front.

The beacons were placed in three iterations. The Figure 4 refers only to the *third iteration* with five beacons. In the first iteration, only one beacon was placed close to the second door. Neither phone nor tablet captured any data in position (F)ront or (B)ehind the tram. At the position (4) and (5) the tablet had weak signals. At positions (1), (2) and (3) phone and tablet were able to receive and verify the data.

For the second run, another beacon was attached just aft the third door and one above the fifth door. This greatly improved reception in the rear areas of the tram and behind. Though due to being positioned to the right side of the body, the phone still had no acceptable readings at positions (F) and (4). In the third run one more beacon was positioned just behind the fourth door and in the front above the right window of the drivers cabin. Now with all five beacons sending as displayed in Figure 4, receiving concurrent messages and verifying them was possible in all positions with the hand held tablet and the phone in the pocket. The pie graphs in the overview fill a quarter for each beacon received. A green quarter relates to complete signature verification within max. 3 seconds. A yellow quarter denotes a varying signal and a verification time up to 6 seconds due to elongated packet capture. In the third run, the lap was extended to the "opposite track's platform" at positions (O).

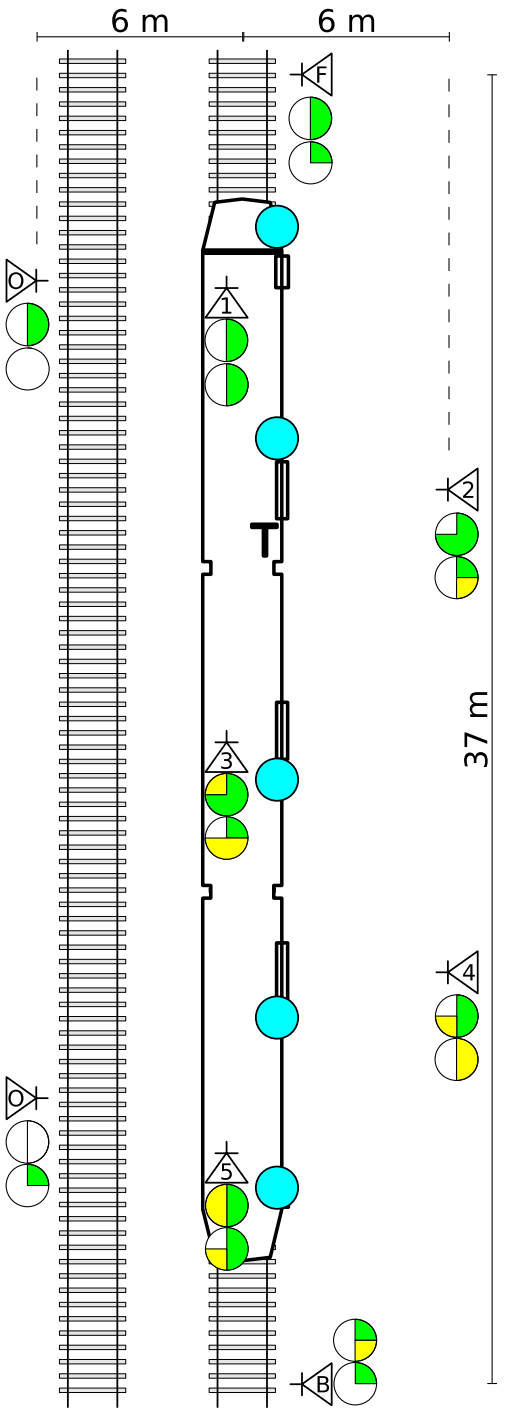

Figure 4. Measurement points (triangles) within and outside the tram. The associated pie graphs (upper: tablet, lower: phone) show the number of well (green) and acceptable (yellow) received beacons with a quarter per beacon.

VI. Conclusion

The presented paper has shown a solution to secure dynamic data broadcast via Bluetooth Low Energy beacons in a many-to-many use case. The concept is compared to existing and even novel published solutions. In comparison, this approach does not rely on concurrent availability of the Internet or a cloud service for data verification. Missing continuous cloud service requests, user tracking is not an inherent "feature". The data broadcast from the original appliance is available to the user instantly also in crowded situations.

The specific use case of a data beacon for public transportation vehicles extends the overall goal to provide door-to-door navigation to passengers. This is especially useful to infrequent travellers, foreign guests and passengers with special needs of accessibility (PRM). Adding security/authenticity is an indispensable key to a dependable system. Though to avoid misunderstanding, complete route navigation still relies on a larger database of additional knowledge available by cached online requests or downloaded to the user device.

The prototype hardware is of an extended beacon with additional data-bus interfaces. In larger productions such a smart beacon would still be quite affordable for mass-adoption, expected in the lower two digit euro-range.

Measurements on a real tram have indicated that beacons can be safely placed inside cabinets. An adequate number of beacons must be ensured to cover the area within the vehicle and outside on the platform. Assumptions were set, that within at least 5 m of the vehicle and within 10 seconds the user device must come up with a verified advice to the user e.g. whether to board an approaching vehicle. These claims were exceeded in the setup shown in former Section V-A with 5 beacons along a 30 m tram. This was no exhaustive investigation. Well placed beacons may also work with larger gap than 7 m. On the other hand, a crowded rush-hour vehicle may show other requirements.

Further research could trial more elaborate tests in a complete fleet with passengers. This could also help to find an optimum signature segment cycle time (T_{cyc}) to balance between low latency and minimum energy consumption on the user's device.

References

[1] M. Rida, F. Liu, Y. Jadi, A. Algawhari, and A. Askourih, "Indoor location position based on bluetooth signal strength," in *Information Science and Control Engineering (ICISCE), 2015 2nd International Conference on*, April 2015, pp. 769–773.

[2] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *Selected Areas in Communications, IEEE Journal on*, vol. PP, no. 99, pp. 1–1, 2015.

[3] S. Czogalla, O.; Naumann, "Pedestrian guidance for public transport users in indoor stations using smartphones," in *18th International Conference on Intelligent Transportation Systems, Las Palmas*, IEEE. IEEE, September 2015, pp. 11–21.

[4] S. Venzke-Caprarese, "Standortlokalisierung und personalisierte nutzeransprache mittels bluetooth low energy beacons," *Datenschutz und Datensicherheit - DuD*, vol. 38, no. 12, pp. 839–844, 2014. [Online]. Available: http://dx.doi.org/10.1007/s11623-014-0329-9

[5] K. Krieger and M. Weksler, "Generating and using ephemeral identifiers and message integrity codes," May 26 2015, uS Patent 9,043,602. [Online]. Available: http://www.google.com/patents/US9043602

[6] D. J. Bernstein, T. Lange, and P. Schwabe, "The security impact of a new cryptographic library," in *Progress in Cryptology – LATINCRYPT 2012*, ser. Lecture Notes in Computer Science, A. Hevia and G. Neven, Eds., vol. 7533. Springer-Verlag Berlin Heidelberg, 2012, pp. 159–176, document ID: 5f6fc69cc5a319aecba43760c56fab04, http://cryptojedi.org/papers/#coolnacl.

[7] M. Düll, B. Haase, G. Hinterwälder, M. Hutter, C. Paar, A. H. Sánchez, and P. Schwabe, "High-speed curve25519 on 8-bit, 16-bit and 32-bit microcontrollers," *Design, Codes and Cryptography*, 2015, document ID: bd41e6b96370dea91c5858f1b809b581, http://cryptojedi.org/papers/#mu25519.

[8] M. Hutter and P. Schwabe, "NaCl on 8-bit AVR microcontrollers," in *Progress in Cryptology – AFRICACRYPT 2013*, ser. Lecture Notes in Computer Science, A. Youssef and A. Nitaj, Eds., vol. 7918. Springer-Verlag Berlin Heidelberg, 2013, pp. 156–172, document ID: cd4aad485407c33ece17e509622eb554, http://cryptojedi.org/papers/#avrnacl.

[9] "Atmel atecc108a product page, summary," 2015, atmel Corporation. [Online]. Available: http://www.atmel.com/devices/ATECC108A.aspx

[10] T. L. P. S. B.-Y. Y. Daniel J. Bernstein, Niels Duif, "High-speed high-security signatures," *Journal of Cryptographic Engineering*, pp. 77–89, 2012, document ID: a1a62a2f76d23f65d622484ddd09caf8, http://ed25519.cr.yp.to/.