Bellrock—Anonymous Proximity Beacons From Personal Devices

Augustin Zidek
Computer Laboratory
University of Cambridge
Cambridge, UK
augustin@zidek.eu

Shyam Tailor
Computer Laboratory
University of Cambridge
Cambridge, UK
shyamatailor@gmail.com

Robert Harle
Computer Laboratory
University of Cambridge
Cambridge, UK
Robert.Harle@cl.cam.ac.uk

***Abstract*—Proximity beacons provide simple, low-cost location data. However, beacon deployments remain rare. In this paper we introduce Bellrock, a framework that repurposes static personal devices (phones, laptops, etc.) as proximity beacons without revealing the location of the device owners, and provides conventional beacons with access control. This is done by using mutable pseudo-anonymous identifiers that can be unmasked by a cloud service.**

We develop Bellrock as a general framework, describing the repurposing scheme and the anonymisation techniques, before applying it to Bluetooth Low Energy beacons. We implement and demonstrate the scalability of the de-anonymisation server, which uses a series of heuristics. We implement a Bellrock client on Android and demonstrate negligible impact on battery lifetime. We evaluate Bellrock using extensive real-world office worker movements. We find that Bellrock was able to provide proximity locations for 8,542 of the 21,796 failed locations that would have occurred without it. We further find that office workers were in range of one or more of their co-workers over 90% of the time, indicating Bellrock can provide relative proximity information even in the absence of a conventional beacon deployment. Overall, we find that Bellrock is both feasible and practical, providing a beacon deployment where there was none, or supplementing existing deployments.

***Index Terms*—Bluetooth Low Energy, Proximity Beacons, Location Tracking, iBeacons**

I. Introduction

Following decades of research effort, ubiquitous indoor location on a par with outdoor (GNSS) location remains elusive. A recent trend has been towards proximity or *microlocation*, whereby spaces are labelled in such a way that mobile devices can detect their proximity. Many variants on this approach exist, including RFID tags, IR beacons, visible light beacons, and radio beacons (particularly those based on Bluetooth). The core idea is to have some spatially-bounded signal associated with static[1] devices (beacons) distributed at known locations in the environment. Observing a specific beacon signal implies co-location with it. Most schemes assume beacons emit Universally Unique IDentifiers (UUIDs) that act as a key into a local or cloud database to find further details such as beacon location. Beacon-based proximity is a simple and robust way to provide location awareness.

However, there are a number of issues that have prevented even modest deployment outside of dedicated testbeds. These include:

- *Deployment and Maintenance.* Finer grained location means beacons with lower spatial range, which in turn requires a higher density of beacons. Deploying and maintaining a dense set of beacons is not an attractive proposition and there are questions over who should own the devices if it is a public service.
- *Ease of Spoofing.* Beaconing is typically one way: the UUID is broadcast to all in range who are listening. Spoofing a beacon by replaying the beacon message is then trivial. Such spoofing would mean a device may be fooled into incorrectly positioning themselves, potentially triggering an unwanted action (for example, using a beacon as a trigger to unlock a door would be unwise).
- *Lack of Access Control.* A beacon message cannot be restricted to a subset of local listeners. Whilst access to the database of beacon information might be restricted, anyone can create their own database of beacons and infer location. Commercially this is a challenge: a company that pays and maintains beacons in a public area cannot prevent the beacons being mapped and used to power a rival service.

In this paper we introduce *Bellrock*[2], which considers ecosystems of beacons formed from:

- *Dedicated beacons.* These are conventional beacon devices, dedicated to beaconing; are statically deployed to known locations; and are designed to be small, long-lasting and inexpensive. They require centralised deployment and maintenance. Bellrock adds access control and spoof protection to traditional deployments of these devices.
- *Personal beacons.* These are beacons that are created by temporarily repurposing a personal device such as a smartwatch, laptop or smartphone. They are dynamic, potentially moving at any time. Personal devices offer

[1] In some circumstances this model is inverted and the beacon is mobile while the listener is fixed. This is particularly common for delivery logistics applications, where packages are tracked by attaching beacons and a network of listeners then monitors location. The conceptual difference is minimal and the discussion presented here applies equally to the inverted system.

[2] Bell Rock Lighthouse is the oldest surviving sea-washed lighthouse, designed to provide location to ships on the coast of Scotland.

more processing, connectivity and battery power than traditional dedicated beacons. Maintenance is distributed across device owners.

Bellrock is the combination of a scheme for repurposing personal mobile devices such as smartphones and laptops to be personal beacons and an anonymisation scheme to retain the location privacy of their owners and provide a degree of access control to dedicated beacons.

Bellrock beacons (either dedicated or personal) broadcast Anonymous IDs (AIDs) over small ranges. They are free to change their AID at any time, subject to some generation rules. If a personal beacon has both WAN connectivity and a position fix, it forwards these to a third party server. A device seeking localisation information listens to the broadcasts and forwards a list of observed AIDs to the server over a WAN connection. The server implements access control before returning location information, if appropriate and available.

We describe the Bellrock framework and implement it using Bluetooth Low Energy beacons. We assess the practicalities and security of the system and evaluate it using two large datasets of personnel movements within office settings. The remainder of the paper is structured as follows: Section II describes the repurposing scheme; Section III the anonymity techniques; Section IV discusses implementation using a cloud server and Bluetooth Low Energy signals; Section V evaluates components of the implementation and uses the movement datasets to explore how the scheme would work in practice; Section VI discusses related work; and Section VII concludes.

II. Bellrock Repurposing Scheme

The primary concern when personal devices broadcast an identifier is that an arbitrary third party can used it to track the device (and hence owner). Bellrock addresses this by asserting that *personal devices will beacon only when they are stationary* (e.g. charging, on a desk, in a pocket while sat down, etc.). Moving devices simply do *not* beacon. Furthermore a device will adopt a new *Anonymous ID* (AID) when it next becomes stationary to prevent pseudonym linkage. We discuss AID generation in Section III-1.

When beaconing, personal beacons differ from dedicated beacons in one important respect: they do not have a predefined absolute location. We can represent a period of time in the Bellrock system as a directional (possibly disconnected) connectivity graph with beacons as vertices, and edges (*A*, *B*) representing an observation by *A* of *B*—see Figure 1. A subset of these vertices are *anchors*, with known location. This includes all dedicated beacons (D1, D2, D3 in Figure 1) and any personal beacon with an out-of-band location (e.g. a GNSS fix, a WiFi position, recognition of a predefined situation such as attachment to a charging cradle, etc.).

Proximity-based location of a device can be determined by the shortest path from the device's vertex to an anchor. The longer the path, the greater the location uncertainty. We use the term '*n*-hop location' to indicate the path has length *n*—conventional proximity-based systems produce 1-hop locations. In Figure 1 device M1 has a 1-hop location; M2 a 3-hop location and M3 a 2-hop location. Clearly the greater the value of *n*, the less confined the device is geographically.

If the graph is dominated by unanchored personal beacons, Bellrock can still provide relative proximity information (i.e. "*A* is co-located with *B*", rather than "*A* is in room R5"). This enables a subset of location-aware applications, including: *localised messaging*, where we want to message everyone in an area (e.g. train platform); *place recognition*, since many places—workplaces in particular—are characterised by the people using them; *interaction monitoring*, where we might want to monitor infection outbreaks, analyse space usage or just log our interactions; and *virtual leashes*, where we need to monitor the continued proximity of personal items such as wallets, keys and children.

A final consideration of repurposing is that of beacon collisions. Beacon technologies are unlikely to implement channel sensing, meaning two overlapping beacons may interfere. In situations such as conference halls and theatres there could be hundreds of Bellrock devices in range of each other, increasing the probability of collision and lost data. It is therefore important that the beaconing procedure incorporates temporal randomness and beacons remain very short relative to the communication rate of the beacon channel. These requirements apply to beaconing in general, so standard technologies typically implement them, as we will show.

III. Bellrock Anonymity, Security and Privacy

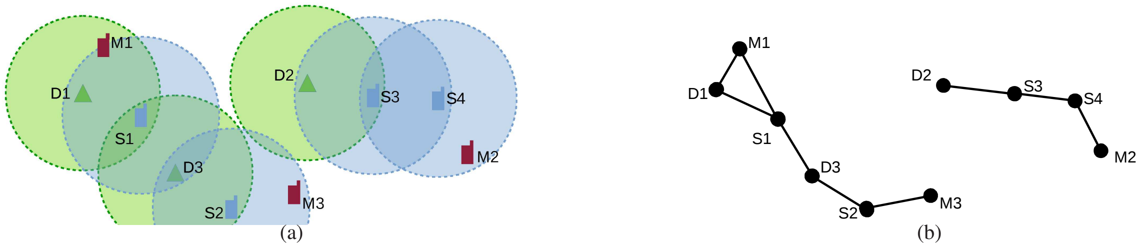
The repurposing scheme must be complemented by an anonymity scheme, or devices will compromise the location privacy of their owners.

1) AID Generation:

Generating pseudo-anonymous identifiers has been studied previously in the passive RFID domain–comprehensive surveys are available in [1], [2]. Many schemes have been proposed, with a variety of trade-offs and security properties [3]. The schemes range from rotating among pre-assigned pseudonyms [4] to detailed cryptographic systems [3]. In principle, many of the general RFID anonymisation schemes can be applied to Bellrock. We advocate a combination of three schemes as follows:

Random AIDs (R-AIDs). Unlike RFID tags, personal beacons may have WAN access, allowing them to negotiate a random AID with a server. This has similarities with RFID schemes that assign and re-assign identifiers from a master reader [4]. It is secure, but the Bellrock AID cannot be changed offline when WAN access is impossible (static beacons) or unavailable (no signal).

Synchronised AIDs (S-AIDs). Google's Eddystone BLE beacon system (developed concurrently with Bellrock) proposes an Ephemeral ID (EID) based on encrypting the current time. The server pre-computes the IDs for all beacons in each epoch and performs a simple lookup for each beacon reported in the epoch [5]. This scheme requires the beacon to have a *synchronised* global clock of sufficient accuracy (a reasonable assumption for personal but not static beacons).



(a) (b)

Fig. 1. Example Bellrock scenario. (a) Devices D1, D2 and D3 are conventional anchored beacons ("dedicated beacons"). S1, S2, S3 and S4 are stationary personal devices acting as personal beacons. M1, M2 and M3 are moving devices that do not beacon. (b) the resultant connectivity graph.

Encrypted AIDs (E-AIDs) Every device, i, has its own unique encryption key,[3] K_i, and generates an AID using encryption: $\mathsf{Enc}_K(\mathsf{Concat}(UUID_i, n))$, where n is a nonce. The server has a list of $\{\text{UUID}_i, K_i\}$ pairs and brute-forces the decryption until the decrypted result is prefixed with UUID$_i$. This allows the device to change its AID at will and without WAN connectivity. It has its roots in the Randomised Hash-Locking scheme for RFID tags [6].

Bellrock uses *symmetric* encryption for E-AIDs since it is significantly faster to decrypt at the server end. We use AES with a block size equivalent to the chosen AID length (i.e. there is only one block) to allow the use of ECB mode and avoid the need to send initialisation vectors.

- *Heuristic 1: Cached Results.* Many people spend the majority of their day in a small number of places surrounded by the same devices and people—for example homes or workplaces). Therefore the same AIDs will be repeatedly observed, often in subsequent scans. To exploit this we:
  (i) Cache recent AID→UUID results
  (ii) Move UUIDs recently observed by the querying device to the top of the search list. Thus a device established as nearby in previous scans but which changes its AID will still be found quickly.
  (iii) Prioritise UUIDs known to be in the same region (within a few hops) of the querying device.
- *Heuristic 2: Friend Priority.* If the cached results do not unmask all the reported AIDs from a device, we then search for the friend and acquaintance devices as a priority. This incorporates the observation that the majority of people we encounter in a typical day are from a small set.
- *Heuristic 3: Spatial Grouping.* Any out-of-band information on location is used to prioritise certain UUIDs. In particular, the serving cell of a smartphone can be matched to those of other devices, suggesting nearby smartphones to prioritise in the search. Databases of cell towers and their locations are readily available (e.g. OpenCellID [7]).

To summarise, Bellrock uses a combination of R-AID, S-AID and E-AID schemes as appropriate. For personal beacons the choice between using S-AID and E-AID will be subtle. In a system of N registered beacons with K AIDs to de-anonymise in an epoch, the S-AID scheme performs N AID generations and k lookups. The E-AID approach will be *maximally bounded* by kN decryptions. However, the heuristics described above should significantly reduce the *effective* N and k. The overall decoding process is:

(i) Check the cache for recent results;
(ii) If no match found, Search the list of R-AIDs for a match;
(iii) If no match found, search the list of S-AIDs generated for the current epoch,
(iv) If no match found, apply the E-AID de-anonymisation process.

2) Adding Access Control with Friends and Acquaintances: When a server has de-anonymised a beacon, it cannot simply report back unmasked UUIDs (or the anonymisation would be redundant). Instead it implements a whitelist of device pairs between which it can share different levels of identity information. The whitelist distinguishes between *friend* and *acquaintance* devices.

If device B is listed as an acquaintance of D, then D is permitted to track B. This means that the server will return a consistent but time-limited pseudonym for B to D. However, it will *not* reveal the true identity of the device or owner. So, for example, a user might be willing to allow a building to track them for safety purposes, but would not wish to divulge *who* they actually are. The friend relationship is used to signal devices to whom the true identity of the device/owner can be revealed. An acquaintance or friend status can be withdrawn at any time by editing the whitelist on the server and changing the device AID.

If two devices are neither friends nor acquaintances, the server can still return anchored location information if it is available. For example it can report a 3-hop location without identifying the intermediate beacons or allowing them to be tracked.

IV. Applying Bellrock to Bluetooth Beacons

So far we have presented Bellrock as a generic proximity beacon framework. In this section we demonstrate its

[3] This is more secure than using a single global encryption key, which makes decryption easy (no key search required) but will inevitably be compromised when the key is extracted and published. An alternative would be the use of asymmetric cryptography, with each beacon having the server's public key. However, this would significantly increase the AID length and the processing cost on the beacon. Google's Eddystone system, which provides a similar beacon anonymisation scheme to Bellrock and explicitly avoids a public key based system for similar reasons.

implementation on the current beacon platform of choice, Bluetooth Low Energy. Bluetooth Low Energy (BLE) is a radio communications protocol designed to maximise battery lifetimes while supporting short Internet-of-Things-like interactions between devices. It was introduced in the Bluetooth 4.0 standard [8] and operates in the 2.4 GHz ISM band at 2400–2483.5 MHz. Forty channels are used with centre frequencies at $2402+2k$ MHz for $k \in \{0,1,\ldots,39\}$. BLE has the notion of *advertisements*, which are short packets of data periodically broadcast at a configurable rate.

For microlocation, dedicated small, inexpensive BLE beacons with long battery lifetimes are fixed within the space of interest. They periodically emit advertisements containing a unique ID over a short range (a few metres). Mobile devices scan for the IDs and use a database to look up the beacon locations and so infer their own context. BLE is a standard inclusion in every 'smart' device today, making it the de-facto choice for location beaconing.

A. Bellrock over BLE

Bellrock uses the BLE advertisement subsystem to broadcast AIDs. BLE advertisement packets are *very* short to prevent the radio subsystem from heating up, allowing simpler componentry. In the most widely-available implementations of BLE (Bluetooth 4.x), the limit is just 31 bytes. The more recent 5.0 standard offers extended advertisements which theoretically provide an extra 256 bytes. However this extended data is provided on on of 37 "secondary" radio channels and not on the same channel as the 31-byte part. This provides backwards compatibility, but adds complexity to e Bluetooth 5 listener. It is expected that small, long-lifetime devices like beacons will *not* use extended advertisements. The current penetration of Bluetooth 4.x radios means any system must retain compatibility with Bluetooth 4.x. Given this we implemented Bellrock for Bluetooth 4.1. The packet format we used required 30 bytes including the 4 byte overhead imposed on the advertisement structure (see the Generic Attribute Profile (GATT) specification [8])[4]:

Len	Type	GAP		AID															
1	2	3	4	5															20

Note that the short packets forced the implementation to use 128-bit AES[5]. This is still sufficiently secure for this application (especially where the keys are changed on a regular basis).

B. Android Client Implementation

We have implemented a Bellrock client for Android smartphones. The client exploits the optimised activity recognition algorithms built into Android to decide when a device should act as a beacon. For a smartphone or tablet, we used the Android `STILL` state to indicate beaconing should begin.

Communication with the server is via a secure HTTPS link using any available WAN connection. Each device can authenticate itself by providing its UUID and current encryption key. Periodically each device sends its list of recently observed AIDs (and timestamps) along with a list of serving cell towers to the server. If the Android location service has a more accurate fix, it also supplies this information.

In implementing Bellrock we found that support for broadcasting BLE advertisements (so-called BLE peripheral mode) is currently patchy. Android as a platform did not support it until version 5.0 (released in 2014). Furthermore, support for the feature in later versions of Android can be disabled by the device manufacturer[6]. Such devices can still act as listeners, able to use the Bellrock framework but not contribute to it.

In terms of beacon collisions: a BLE advertisement lasts up to 0.376 ms (376 bits at 1 Mbps), and incorporates a random jitter of 0–10 ms on each period to ensure two colliding beacons do not continue in lockstep. In a very dense environment of hundreds of Bellrock devices, then, collisions are expected but the majority of beacons will be transmitted. In principle, Bellrock only requires a few successful beacons to be sent each period, provided a different set of beacons is successful in each subsequent periods.

C. Server Implementation

The server process is dominated by the AID decoding task. Our implementation was written in Java 8 and took advantage of AES-NI (Intel CPU instructions for fast AES operations) [9]. Additionally, since initialising a `Cipher` object in Java has a large overhead, the server maintained a pre-initialised `Cipher` object for each user in memory. The OpenCellID database was downloaded, pre-processed and incorporated locally in the server to avoid constant network lookups.

V. Evaluation

A full-scale deployment with thousands or millions of users was not feasible for this work. Instead we focus on evaluating three aspects: 1) the feasibility of the anonymisation scheme at the server; 2) the impact on mobile device lifetimes; and 3) the utility of personal beacons within a proximity deployment.

A. Server Feasibility

The worst case server situation is all devices using the E-AID scheme. We tested the Bellrock server as described in Section IV-C on modest hardware: a 2011 Intel Core i5-2410M mobile CPU running at 2.3 GHz running a Java Virtual machine allocated 3 GB of memory. This was able to perform approximately 5,600,000 AES decryptions per second. The practicality of this number depends on the number of decryptions needed to decode a single AID, which we expect to scale linearly with the number of users in the system. Figure 2 illustrates the results of the time taken to decrypt 1,000 AIDs chosen at random from a set of N users, where

[4] We assumed the use of a registered manufacturer's short ID to avoid having to specify a 16-byte full ID.

[5] We used 64-bit UUIDs and therefore 64-bit nonces.

[6] In our experience BLE peripheral mode was supported by the majority of current Android smartphones, as well as Apple's iOS.

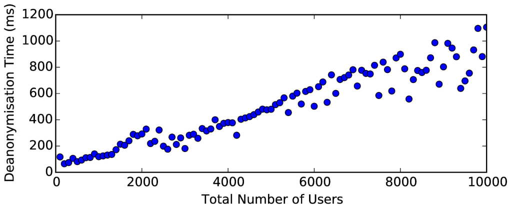


Fig. 2. Time to decrypt 1,000 AIDs with number of users ranging from 100 to 10,000. The AIDs were sampled randomly from all users and no heuristics were applied.

N varied from 100 to 50,000. The linear trend is evident. The time costs are tolerable for a few thousand users but prohibitive when we reach tens of thousands. A global deployment could see many millions of users and this is clearly untenable.

To scale to a reasonable number of users requires that the heuristics keep the search space small (up to a few thousand). The serving cell spatial heuristic is particularly effective here. Based on the numbers presented in [10], which considered the deployments of two large national cellular service providers, base stations are deployed such that each serves around 1,000-2,000 users[7] A search for a new AID should then only involve searching a list of a few thousand. Without any other heuristics in place, it would therefore be necessary to make around 1,000 decryptions per AID on average. This suggests our test server would be expected to decode around 5,000 AIDs per second.

This is a pessimistic estimate given that the test server used modest hardware, did not consider distributed computing (the process is inherently parallelisable) and ignored the other de-anonymisation heuristics. In fact the AID→UUID caching is likely to be the key factor. Beacon transmission ranges are of the order of metres and people density is relatively low. As we show shortly, we only expect someone to be in range of a small number of others (fewer than ten). There are corner cases where this is not so (commuter trains, lecture theatres, etc.) but such circumstances are in the minority on the whole and we can tolerate lengthier decryptions for those rare cases. Overall, then, the server de-anonymisation is a realistic proposition on today's hardware.

B. Android Validation

The Android implementation was tested on a number of consumer Android devices with Android versions of 5.0 or greater. We verified that the built-in activity recognition was sufficient to recognise static periods with high confidence; and that BLE beaconing only occurred during those periods.

To assess Bellrock's battery impact we used Android's BatteryStats tools [11] applied to a Google Nexus9 device running stock Android 7.1.1 with a test app installed. We ran a series of three-hour tests with WiFi disabled. This duration spanned the various sleep states Android has (power consumption is reduced in stages, with a deep sleep after an hour). We whitelisted the app to prevent Android from altering its behaviour to save power.

Test	Description	Draw (mAh)
Control	No apps running	7.7/6.0/6.0
Control + wake-lock	Test app just holds a partial wakelock	38.3/31.4/27.9
BLE Beaconing	Beaconing every 100 ms, no wakelock	26.4/6.0/6.0
BLE Beaconing + wakelock	Beaconing every 100 ms with partial wakelock	45.5/33.2/34.4
BLE Scanning + wakelock	Scanning continuously next to device beaconing at 10 Hz	70.3/65/65
E-AID Generation + wakelock	Generating E-AIDs continuously every 10 ms	47.8/42/43.7

TABLE I
ANDROID ENERGY CONSUMPTION TESTS.

The tests and results are described in Table I: the mean consumption values for each hour are given. The second and third hour results represent the consumption with the system in full 'doze' mode. We observe that the cost of BLE beaconing (which was at full power and 10 Hz) was small in the first hour (around 20 mAh or less than 1% of a typical 3,000 mAh smartphone battery) and negligible thereafter (we verified that the beacons were still sent using a nearby listener). This is expected: BLE was designed for exactly this type of scenario and the Bluetooth subsystem can beacon without CPU involvement. So the steady state cost is little more than the cost of powering the Bluetooth chip.

For the AID generation test over 350,000 were generated in an hour for a total cost of around 45 mAh. As we demonstrate shortly, we expect a device to use only tens of AIDs per hour, making this cost equally negligible. Scanning was a more costly process (60–70 mAh per hour), but we only expect a device to scan intermittently so the real-world cost is unlikely to be of significance to the average user. Overall, then, we consider Bellrock to have minimal impact on a phone's battery lifetime.

C. Evaluation Data

To explore the utility of personal beacons we used two established datasets recording in-situ movement data within office environments. Bellrock is well suited to this environment: employees are often stationary, bring various personal devices with them, and we might expect them to be in range of at least one other most of the time. The datasets were:

- *SpaceLab BLE Data.* This dataset was collected over four weeks in an architectural company (SpaceLab Ltd.) with a building spanning two floors. The data are derived from 25 employees who wore a custom watch-like device that acted as a BLE beacon whilst continuously scanning for other BLE devices. Full details of the experimental setup are given in [12].
- *AT&T Research Movement Data.* This dataset was collected at the former AT&T Research Laboratories in Cambridge, UK and contains in-situ measurements of 57 employees. They were tracked to within a few centimetres using an ultrasonic location system installed throughout

[7] This estimate is derived by dividing the subscriber numbers in [10] by the base station numbers.

the three-floor laboratory [13]. The logs span 139 days over a two year period in the early 2000s, with a total of 22,667,607 location results. Note this dataset does not contain any beacon data, Bluetooth or otherwise.

1) Data Preprocessing: Both datasets underwent cleaning before usage. The SpaceLab data contained the output of a walk detection algorithm on the device. For any period where steps were observed for a device, we discarded all observations of it since the Bellrock scheme would not have let it beacon. The data included observations of a set of dedicated beacons. Any device that did not observe a dedicated beacon for one minute was assumed to have left the building.

The AT&T data did not contain any beaconing data so we synthesised the output of a beacon as a simple radius. Any two people within 7.5 m of each other were assumed to be able to observe each other. Movement in the AT&T was easily inferred from the high accuracy positions. For each device, a day was defined as starting with the first recorded walk and finishing with the last.

2) Beaconing Statistics: Figure 3 shows the distributions of the total time each device would have beaconed for in each day for both datasets—this is equivalent to the time the owner was active but *not* walking. The numbers represent a real working environment and do not compensate for the number of hours worked. The two distributions are very similar. The median beaconing times were 28,418 s (SpaceLab) and 26,925 s (AT&T)—a high proportion of a typical working day. This is to be expected of an office environment, where employees spend much of the time in a sedentary state. This is promising for Bellrock, which depends on high proportion of personal devices beaconing at any given time.

3) AID Lifetimes: Although the total time spent beaconing is a significant proportion of a day, it is composed of many small periods of continuous beaconing punctuated by movement activity: a meeting, going to lunch, or a trip to the kitchen or bathroom. Figure 4 shows the distribution of these beacon periods across all devices and all days in the datasets (note the logarithmic scales). Once again the shapes of the two distributions are more similar than different[8]. The beaconing periods are typically anywhere from a few seconds to a few minutes. The shorter times are associated with small movements at a desk or short pauses when walking. The SpaceLab distribution hints at a more active set of employees, but this is more likely due to the SpaceLab measurement device being worn on the wrist and the activity recognition producing false positives due to arm movements. The long tails on these distributions are significant: in the AT&T data, over 23.8% of AIDs would have lived longer than 10 minutes

In the Bellrock scheme, each of these periods is associated with a distinct AID so Figure 4 is equivalently a view of the AID lifetime distribution. A given device therefore changes its AID multiple times during a typical day—Figure 5 shows the distributions for the two datasets. Here we notice a significant difference in shape, with the AT&T simulation producing fewer AID changes per hours (median 4.0 vs 21.3). We attribute this to the higher false positive mobility of the SpaceLab participants due to measurement on the wrist, although differences in mobility pattern may also have contributed. We note that the typical number of AID changes per hour means that each AID will span many observation periods and that the server will find the AID in its cache more often than not.

D. Positioning Results

1) Mixed Dedicated/Personal Beacon Deployment: We use the SpaceLab dataset to demonstrate the value of Bellrock. In addition to the wearable beacons, this dataset contained 17 dedicated beacons, deployed to give approximately complete coverage of the building. In such systems there are inevitably location blackspots, where dedicated beacons cannot be reliably observed (i.e. no 1-hop location). We searched each 5 s period of each day for such situations.

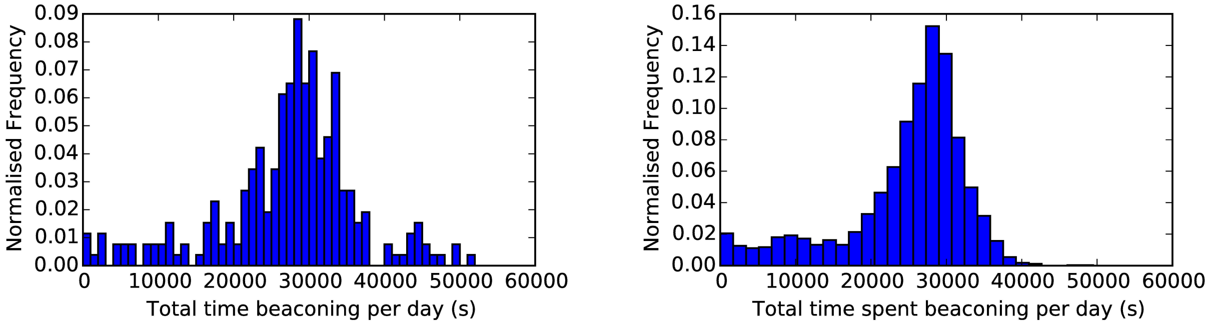
In these cases we constructed a directional graph as per Section II, with a small modification. We established a single 'anchor' vertex and assigned all dedicated beacons to that vertex. We then found the shortest path from the device with no 1-hop location to the anchor node. Figure 6(a) illustrates the results. *Of 21,796 positions that would have failed, Bellrock was able to recover a position for 8,701 (40.3%).* Of these 8,542 (98.2%) were 2-hop locations, so the location error would still have been well bounded.

The 17 dedicated beacons within the SpaceLab data were deployed in an attempt to give near-complete coverage. To further demonstrate the value of Bellrock for location we reprocessed the data 29 times, each time deleting one or more dedicated beacons. Due to the large search space we sampled the beacons at random when the number of beacons to be removed was two or greater. Figure 6(b) illustrates the results, where white segments indicate failed locations that Bellrock could not assist with; blue segments where Bellrock was able to recover a 2-hop location, etc. There are two clear trends: the first is that Bellrock was able to recover around half of the positions in every case, the vast majority of which were 2-hop locations. Combining all of the test results together, of 982282 failed 1-hop locations, Bellrock recovered 507255 (51.6%) as 2-hop, 11848 (1.2%) as 3-hop and 262 (0.02%) as 4-hop locations. The second trend is that the total number of failures increases as the number of beacons removed increases. This is to be expected—fewer beacons give less coverage in general.

We conclude from these results that personal beacons can significantly increase robustness within a dedicated beacon deployment.

2) Scope for Personal Beacon-Only Deployments: The majority of spaces today have no dedicated beacon deployment for Bellrock to enhance. It is interesting to understand the extent to which Bellrock could provide useful location data in such circumstances. Clearly the answer is dependent on many factors including the number of users in an area, their

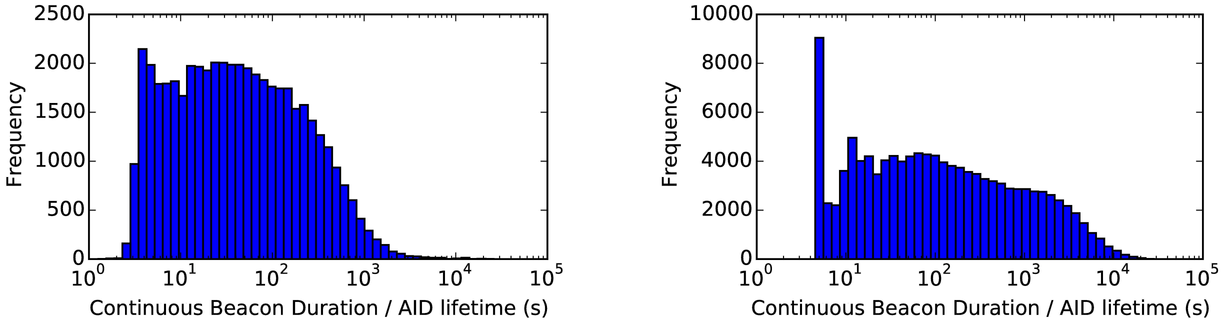
[8] The large peak in the AT&T data at 5 s and the lack of data less than 5 s is an artefact of the Bellrock simulation, which had devices beacon every 5 s. Thus this spike should arguably be distributed over the range 0–5 s.



(a) SpaceLab data

(b) AT&T data

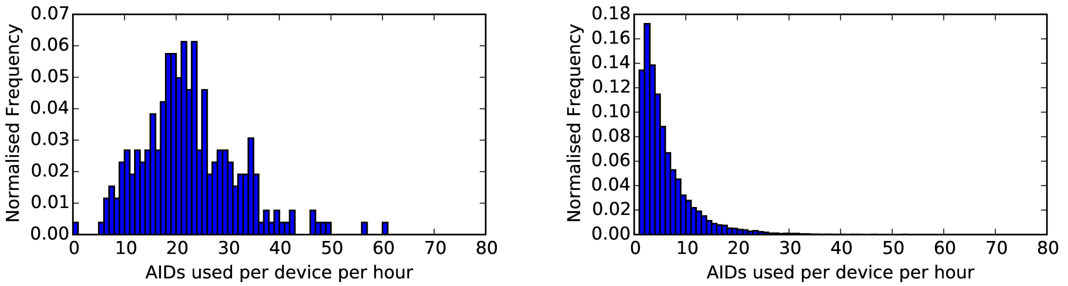
Fig. 3. Histogram of the total time spent beaconing by each device for each day



(a) SpaceLab data

(b) AT&T data

Fig. 4. Histogram of the beacon period durations across all devices and days. (Note the logarithmic x-axes).



(a) SpaceLab data

(b) AT&T data

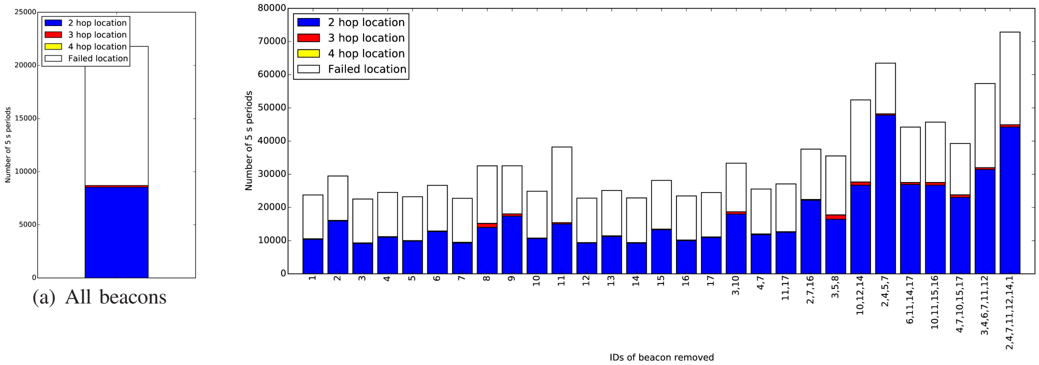
Fig. 5. Histogram of the number of AID changes per hour, taken across all devices on all days.

mobility patterns, the environment layout, the beacon range, etc. Nonetheless, we can use our datasets to assess sample office environments. Arguably such an environment is the best case: lots of personal devices in close proximity with owners who are predominantly sedentary.

Without dedicated beacons Bellrock relies on being able to hear at least one other personal beacon, and preferably more. Figure 7 gives a histogram of the number of beacons that could be heard over 5 s intervals in the two datasets. The key observation from is that, despite the different sources, the distributions in Figure 7 are very similar, with both showing peaks around 5 neighbours. Most importantly the proportion of time spent out of range of any other beacon was less than 10% (AT&T data) or 6% (SpaceLab data). Thus Bellrock could have provided proximity information (albeit most likely relative) over 90% of the time.

E. Security and Privacy Evaluation

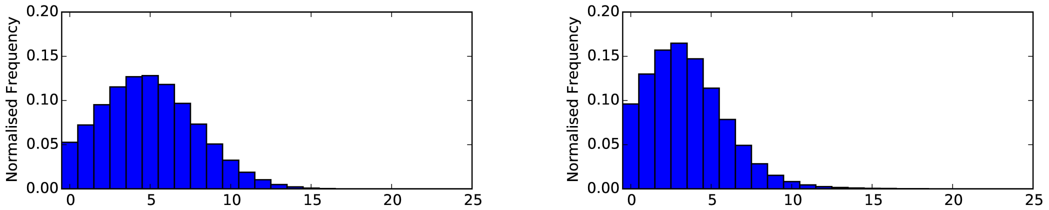
As with many systems, there are a few pathological conditions under which Bellrock is vulnerable. To explore these, we first make the following assumptions: the Bellrock server is secure, i.e. adversaries have no access to its databases, binaries



(a) All beacons

(b) Beacons removed

Fig. 6. SpaceLab proximity location results



(a) SpaceLab data

(b) AT&T data

Fig. 7. The distribution of number of neighbours

or server memory; Bellrock client-server communications are secure; and there is access to a source of random numbers with high entropy. All of these assumptions are achievable using standard architectures.

Spoofing. Bellrock can reduce the risk of spoofing by regularly changing the AID. For the E-AID scheme, a counter would need to be incorporated in the AID generation and the server would need to reject any device with a counter lower than the highest observed.

Denial of Service (DoS). Bellrock is vulnerable to various Denial-of-Service (DoS) attacks. Most beaconing technologies will be susceptible to jamming of the beaconing channel. However, this is typically highly localised and unlikely to benefit the attacker. More likely is an attack on the server by overloading its capacity. This could be achieved by, say, reporting artificial beacon messages containing random AIDs. Bellrock addresses this by requiring all devices to have a UUID and encryption key issued by the server, even if they never beacon. When reporting sightings, device use this pair to identify themselves. The server can then blacklist devices that repeatedly submit unresolvable AIDs.

Brute Force Key Search. An attacker could perform a brute-force decryption to find the key of a device. They would need to observe multiple AIDs from the same device and brute-force the decryption, looking for outputs where the first eight bytes (the UUID) matched. This is, however, very costly—generously assuming the attacker can use a large network of machines to try 10^{12} keys per second[9], it would still take about $\frac{2^{128}}{10^{12}} \approx 10^{29}$ s which is approximately 10^8 times the age of the universe assuming AES with 128-bit keys. Even if the attack succeeded, the attacker has only found the key for one device and cannot decode any others. Furthermore, regularly changing device keys when they communicate with the server minimises the impact of such an attack.

Low Node Density. When there are few devices operating in an area, Bellrock cannot prevent AID-linkage and hence tracking. As an example, consider a lone user in a building. Regardless of the AID being broadcast, they can be trivially tracked since they are the only beacon present. A related problem arises when significant spaces are owned by a single person (e.g. a single-occupancy office). In these cases the system is vulnerable: however, there are many other ways to track individuals in this situation, such as WiFi tracking.

Power Matching. If a beacon changes its AID without a period of movement, the change may be observable from a static listener. This listener would see one AID disappear, only for another to appear with very similar characteristics (RSSI values, etc). For this reason it may be sensible to vary the transmit power alongside the AID in such a way that the server knows the current transmit power but a listener does not. Similarly, a randomised pause between AID changes would be advisable.

VI. Related Work

Bluetooth Beacons. Beacons based on Bluetooth Low

[9] Our benchmarks had one machine decrypting 10^6 keys per second.

Energy are a popular way to provide proximity location due the wide availability of the technology. Apple pioneered the use of BLE beacons through their iBeacon product [14], [15], [16] and other manufacturers have followed with other variants [17], [18]. All are layered on the core BLE protocol, making them more similar than different. their use for location services has triggered privacy concerns ([19], [20], [21]), motivating anonymous beaconing.

Anonymous Beaconing. As outlined in Section III-1, anonymous beaconing has been studied extensively in the RFID domain [1], [2]. In principle any anonymisation scheme can be used by Bellrock.

Anonymous BLE beacons are also commercially available, although details are scant. All of the schemes involve a trusted third party as per Bellrock. Estimote's *Secure UUID* [22] system assigns each beacon a unique key at manufacture, used to generate a *visible ID* to broadcast. An observer of the Visible ID forwards it to a cloud server to obtain the real ID, but further details are not published. In parallel with this work, Google have released the Eddystone BLE beacon platform [18], [5], which offers beacons an ephemeral ID (EID) that changes every few minutes. The EID can be resolved to a unique ID using a trusted third party. Although developed independently, the Eddystone EID is equivalent to the Bellrock S-AID scheme, involving the third party pre-computing new EIDs at the start of each epoch and then performing lookups.

Note that the focus for BLE to date has been on anonymising a manual deployment of static beacons for managed environments such as retail stores. Bellrock aims to provide location services both to enhance and to knit together these small, dedicated deployments.

Crowdsourcing. Bellrock's repurposing scheme is close in nature to "crowd-GPS" systems such at Tile [23] or TrackR [24]. These use tags (usually BLE-based) attached to important devices such as keys. Smartphones act as observers, running a client-side app that searches for tags whenever the phone knows its coarse location. and reporting a tag sighting to a central server. The owner of a lost tag can query the server to find the last known sighting. Commercial systems currently have very weak privacy guards, prompting the semi-decentralised Techu system [25]. In Techu, observers report a triple of {timestamp, observed tag ID, pseudonym} to an *untrusted* server. The pseudonym is a handle that can be used to connect to the device. To find a tag, a device retrieves this pseudonym from the server and uses it to contact the observing device. It then proves ownership of the tag and is told the location that the device observed it.

The design decisions for Techu reflect its usage model: irregular, infrequent requests for a coarse (perhaps building-level) location of a tag. By contrast, Bellrock is associated with regular requests for more accurate location information. Both systems have a single point of failure in the form of a central server, but a compromised Techu server will not reveal any private information, unlike the Bellrock server. A Techu-based Bellrock system would have personal devices both beaconing and continuously listening, reporting back their observations to the server. A device seeking its location would provide its observations to the server and receive back a list of communication IDs. It would then need to contact the devices to ask if they had a location fix. Unfortunately this is not practical, partly because of the significantly increased power drain associated with constant listening; partly because it assumes a WAN is always present; and partly because static beacons will not have the power or communications hardware to participate.

Location for Ad-hoc and IoT Networks. The localisation of nodes from connectivity graphs has been studied within the context of Ad-hoc wireless networks. Here, nodes distributed arbitrarily in space must localise themselves, either relatively or absolutely [26], [27], [28], [29]. Typically the techniques require some form of accurate distance or angle sensor, but in principle could extend Bellrock significantly.

VII. Conclusions and Further Work

In this paper we have described the Bellrock framework, designed to give conventional dedicated (non-personal) beacons access control and allow the use of personal device as anonymous proximity beacons to improve (or provide) beacon coverage. This is achieved by permitting beaconing only when the device is stationary and ensuring the content of the beacon is a mutable pseudo-anonymous ID that can be de-anonymised using a cloud-based service. We have described the implementation of the framework using Bluetooth Low Energy (BLE) advertisements on Android, finding minimal battery impact for users.

We have evaluated Bellrock using two extensive datasets of office personnel movements, one of which collected BLE advertisements between wrist-worn beacons/listeners and static dedicated beacons. For that dataset we found that Bellrock was able to provide locations for 8,542 of the 21,796 failed locations that would have occurred in the system. When the dedicated beacon density was reduced, Bellrock performed similarly, providing locations for over 50% of the failed locations. Our analysis found that office workers were out of range of their colleagues for only 10% of the time, indicating Bellrock can provide relative proximity-based locations 90% of the time, even without a dedicated beacon deployment.

In future work we intend to explore the extent to which the spatial geometry of the beacon network can be recovered, applying localisation techniques from wireless ad-hoc networks. We also intend to explore the use of more advanced positioning methods such as radio fingerprinting [30]. Finally, we hope to investigate ways to beacon while moving without adversely impacting privacy.

VIII. Acknowledgements

The authors would like to thank the reviewers for their detailed and insightful comments.

References

[1] A. Juels, “RFID security and privacy: a research survey,” *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 381–394, Feb 2006.

[2] S. Piramuthu, “Lightweight cryptographic authentication in passive rfid-tagged systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, pp. 360–376, May 2008.

[3] Z. Shi, S. Ren, F. Wu, and C. Wang, “The vulnerability analysis of some typical hash-based rfid authentication protocols,” vol. 04, pp. 1–9, 01 2016.

[4] A. Juels, *Minimalist Cryptography for Low-Cost RFID Tags (Extended Abstract)*, pp. 149–164. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[5] A. Hassidim, Y. Matias, M. Yung, and A. Ziv, “Ephemeral identifiers: Mitigating tracking & spoofing threats to ble beacons,” tech. rep., Google, 2016.

[6] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, *Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems*, pp. 201–212. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[7] “OpenCellID.” http://opencellid.org, 2018. [Online; accessed January 2018].

[8] S. Bluetooth, “Bluetooth specification version 4.2,” *Bluetooth SIG*, 2014.

[9] N. Firasta, M. Buxton, P. Jinbo, K. Nasri, and S. Kuo, “Intel AVX: New frontiers in performance improvements and energy efficiency,” *Intel white paper*, 2008.

[10] L. Chiaraviglio, F. Cuomo, M. Maisto, A. Gigli, J. Lorincz, Y. Zhou, Z. Zhao, C. Qi, and H. Zhang, “What is the best spatial distribution to model base station density? a deep dive into two european mobile networks,” *IEEE Access*, vol. 4, pp. 1434–1443, 2016.

[11] “Android battery historian.” https://developer.android.com/studio/profile/battery-historian.html, 2018. [Online; accessed January 2018].

[12] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer, “A study of bluetooth low energy performance for human proximity detection in the workplace,” in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 90–99, March 2017.

[13] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, “The anatomy of a context-aware application,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom ’99, (New York, NY, USA), pp. 59–68, ACM, 1999.

[14] M. Köhne and J. Sieck, “Location-based services with ibeacon technology,” in *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*, pp. 315–321, Nov 2014.

[15] T. Ming Ng, “From "Where I Am" to "Here I Am": accuracy study on location-based services with IBeacon Technology,” *HKIE Transactions*, vol. 22, 04 2015.

[16] M. S. Gast, *Building applications with IBeacon: proximity and location services with bluetooth low energy*. " O’Reilly Media, Inc.", 2014.

[17] “Estimote.” http://www.estimote.org/, 2018. [Online; accessed January 2018].

[18] “Google eddystone.” https://developers.google.com/beacons/eddystone, 2018. [Online; accessed January 2018].

[19] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno, “Devices that tell on you: Privacy trends in consumer ubiquitous computing,” in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS’07, (Berkeley, CA, USA), pp. 5:1–5:16, USENIX Association, 2007.

[20] W. FL. and S. F., “Location privacy in bluetooth,” in *Security and Privacy in Ad-hoc and Sensor Networks. ESAS 2005. Lecture Notes in Computer Science vol 3813*, Springer, Berlin, Heidelberg, 2005.

[21] H. Kikuchi and T. Yokomizo, “Location privacy vulnerable from bluetooth devices,” in *2013 16th International Conference on Network-Based Information Systems*, pp. 534–538, Sept 2013.

[22] “Estimote Secure UUID.” http://developer.estimote.com/ibeacon/secure-uuid/, 2018. [Online; accessed January 2018].

[23] “Tile.” https://www.thetileapp.com/, 2018. [Online; accessed January 2018].

[24] “Trackr.” https://secure.thetrackr.com/, 2018. [Online; accessed January 2018].

[25] I. Agadakos, J. Polakis, and G. Portokalidis, “Techu: Open and privacy-preserving crowdsourced gps for the masses,” in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys ’17, (New York, NY, USA), pp. 475–487, ACM, 2017.

[26] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom ’01, (New York, NY, USA), pp. 166–179, ACM, 2001.

[27] L. Doherty, L. El Ghaoui, *et al.*, “Convex position estimation in wireless sensor networks,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1655–1663, IEEE, 2001.

[28] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, MobiCom ’03, (New York, NY, USA), pp. 81–95, ACM, 2003.

[29] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 22, pp. 54–69, July 2005.

[30] R. Faragher and R. Harle, “Location fingerprinting with bluetooth low energy beacons,” *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 2418–2428, Nov 2015.