UNIVERSITÉ DE GENÈVE

Archive ouverte UNIGE

https://archive-ouverte.unige.ch

Chapitre d'actes 2016 Accepted version Open Access

This is an author manuscript post-peer-reviewing (accepted version) of the original publication. The layout of the published version may differ .

Beacon Authpath: Augmented Human Path Authentication

Huseynov, Emin; Seigneur, Jean-Marc

How to cite

HUSEYNOV, Emin, SEIGNEUR, Jean-Marc. Beacon Authpath: Augmented Human Path Authentication. In: Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence. [s.l.] : IEEE, 2016.

This publication URL: https://archive-ouverte.unige.ch/unige:91452

© This document is protected by copyright. Please refer to copyright holder(s) for terms of use.

Beacon AuthPath

Augmented Human Path Authentication

Emin Huseynov
Faculté des Sciences de la Société
University of Geneva
Geneva, Switzerland
emin@huseynov.com

Jean-Marc Seigneur
CUI, ISS & Medi@LAB, Faculté des Sciences de la Société
University of Geneva
Geneva, Switzerland
seigneuj@gmail.com

***Abstract*— BLE (Bluetooth Low Energy) beacons are being deployed in smart cities, especially to augment the shopping experience of customers in real outlets. Thus, humans as they walk in cities pass by different beacons and the sequence of beacons form a path. In this paper, we present how those augmented paths can authenticate a user in a secure way whereby the users can prove they have passed by a path, even with current unsecure beacons. We have built a prototype to validate this new authentication scheme with unsecure Estimote beacons. In addition, as an alternative to BLE, a similar system utilizing Wi-Fi to detect user proximity is reviewed as well.**

***Index Terms*— Authentication, Physical security, Augmented Reality.**

I. Introduction

As per Stajano [1] and Dey [13], ubiquitous computing with all its sensors embedded in the environment and carried out by humans has open the door for novel context-aware authentication schemes. Determining exact geographic location of a person is always seen as a way of proving identity. While there are obvious ways of implementing this, such as determining coordinates using a space-based system, such as Global Positioning System (GPS) or getting location information based on Internet connection used, they are not accurate and trustworthy: IP address based information's accuracy is generally limited to determining a city/village/neighborhood, GPS's can only be used outdoors and in areas with direct satellite coverage. Furthermore, both methods are relatively easy to forge, jam or spoof.

In this paper, we present how augmented paths can authenticate a user in a secure way whereby the users can prove they have passed by a path, even with current unsecure beacons. In Section II, we survey related work. Section III details our new scheme: its model, use-cases and alternative implementations. In Section IV, we present how we have validated this new authentication scheme with unsecure Estimote beacons.

As the technology is based on the new Bluetooth standard (4.0), older devices will not be supported [3]. As an alternative, in one of the use-cases, we will replace BLE based broadcast with Wi-Fi SSID based broadcasts. This will help overcome the limitations of Bluetooth 4.0.

II. RELATED WORK

Electronic geo-fencing [2] is a technique that has been proposed to ensure that people, devices and machinery are accessed in or from authorized physical locations only. In contrast, our scheme focuses on secure paths with currently deployed BLE technology.

Authentication using a virtual iBeacon as a second factor has been used in the product offered by SAASPASS [15]- the application installed on a user's smartphone automatically transmits the generated onetime password to a special connector (a BLE listener daemon- currently only available for MacOSX). This connector searches for BLE packets transmitted in the near or immediate range and if the packet parameters match, passes the values gathered to the application (e.g. a browser) emulating keyboard keys.

The same idea, but using hardware BLE Beacons has been researched by van Rijswijk-Deij [16]. The main concept is to replace the static BLE device identifiers by dynamically changing attributes that can serve as one-time passwords to be used for authentication.

IT security is not the only area where BLE beacon technology may appear; Bluesmart [12], marketed as a "smart suitcase" relies on BLE beacon technology for physical security: the owner can open the suitcase by just approaching or tapping on an app installed.

Although the use Wi-Fi as a proximity base was already researched [17], transmitting OTPs as a part of Wi-Fi SSID seems to be a new idea; we were not able to identify any related or similar work in this area.

III. BEACON AUTHPATH MODEL

In this section, we describe our new Beacon AuthPath model including two use-cases. The model is based on authenticating a user's physical path by checking proximity to a set of geographic locations (checkpoints) in a predefined order with proximity distance varying depending on technology used. When the user with a mobile device running the AuthPath application approaches a checkpoint, the application records various parameters, such as current time and the data broadcasted by the beacon device (beacon data) the checkpoint is equipped with. The same procedure repeats on with other checkpoints the path consists of. This data then is transferred to a validation server, which checks the submitted data using predefined algorithms and accepts or denies the person as an authenticated user.

The first case is based on a system with standard beacons; other two use-cases will utilize enhanced beacons, called “smart beacons”, based on BLE and Wi-Fi SSID broadcast to transmit the security context information to the mobile application.

The security level of the first use-case is low as the beacon data for all checkpoints are static and therefore the use-case is vulnerable to replay attacks: the beacon data may be passed from a user physically present in the checkpoint to any remote user, allowing the latter to forge the authentication path. In the same time, the low cost and availability of standard beacon devices make this use case very easy to implement, therefore this use-case may still be of an interest in scenarios where low security level is tolerable (e.g. if authentication path is used as a second factor or in augmented sports etc.)

The other two use cases use dynamic beacon data, periodically changing similar to one-time password mechanisms, therefore the attack window per every checkpoint is very narrow. With a large number of checkpoints used in an authentication path and very short beacon data rotation period (even 1 second is possible if system times of all devices are correctly synchronized), the replay attacks become practically impossible.

A. *“Beacon AuthPath” with standard beacons*

1) Standard BLE-based beacons

One of the possible use-cases can be a system for verifying a path passed by a person within a wider geographical area, such as a smart city. Standard BLE beacons transmit radio packets consisting of four main pieces of information: “UUID” – a 16-byte unique identifier of the device, “Major” and “Minor” - 2-byte string and “TX Power” – value used to determine the distance between devices [3].

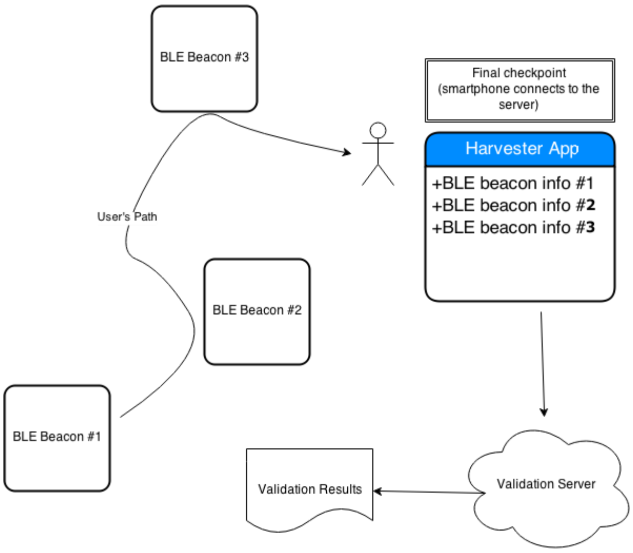


Fig. 1. AuthPath sequence diagram

The mobile application used in this use case will detect this information, save locally together with current timestamp, and transfer the harvested data to the server at the final checkpoint. To protect from the replay attack (e.g. generating a virtual beacon with the same parameters) the mobile device then initiates changing the value of “Major” variable of the BLE device to a random value using a special SDK built in the application. The app sends the new value to the server after it assigning it to the beacon device. However, this is only possible if the SDK of BLE device allows such manipulation; the availability and mechanism of this feature is purely on manufacturer’s’ discretion and is not defined in iBeacon or any other specifications.

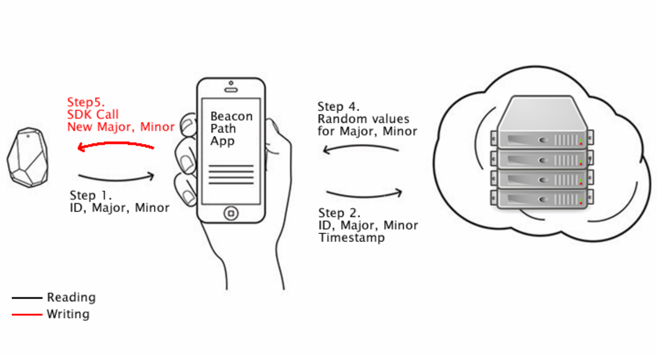


Fig. 2. Beacon AuthPath data flow

This use case makes use of standalone BLE beacons that can be managed via Bluetooth using an SDK call, and with no security mechanisms in place. Estimote beacon [4] is an example of such device and is planned to be used in the proof-of-concept implementation.

2) Security analysis

The model with standard beacons provides limited security, as the beacon broadcasts are publicly visible. With the scenario where the app uses SDK calls to modify beacon’s parameters, the risks of “replay” attacks are minimal (as the server generates and sends new values only to the last user passing the checkpoint). However, as the Estimote’s SDK provides no protection, the authentication sequence is insecure. This is a major obstacle to using the given scenario in systems requiring higher level of security.

B. *“Beacon AuthPath” with “smart” beacons*

1) BLE-based “Smart Beacons”

This is a slight modification of the previous use case where standard beacon devices are replaced with compact devices (“smart” beacons) running an operating system capable to periodically run simple scripts. This will allow avoiding modifying beacon values when passing the checkpoints; instead, the major value will be regenerated automatically every N seconds. This also removes the requirement of the mobile device to be online, the collected beacon information can be stored locally and uploaded to the server in bulk at any convenient time or place, e.g. when reaching the final destination point. Each “smart” beacon will have a secret hash key (not visible, stored in device’s internal memory). The Major value broadcasted by smart beacon device is generated using TOTP [5] algorithm based on current time and the secret hash key. When user approaches the beacon, the broadcasted packet information (UUID and Major) is saved to mobile devices' memory together with current timestamp. After the path is completed, the array of information gathered at all checkpoints and afterwards uploaded to the server. The server will have a copy of hash keys of all smart beacons and is able to analyse the submitted matrix and verify whether it contains valid "Major" values, by rerunning TOTP algorithms with the submitted timestamps and stored hash keys, and comparing with the submitted "Major" values.

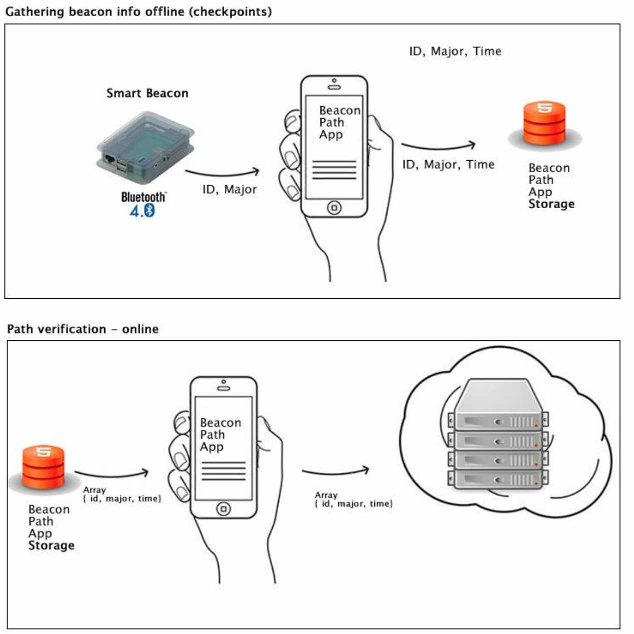


Fig. 3. Beacon AuthPath with smart beacons data flow (online and offline)

2) *Wi-Fi SSID based "Smart Beacons"*

This use-case is similar to BLE Based "Smart Beacons" but uses Wi-Fi SSID as the mean of transmitting the broadcasts from the smart beacon to the mobile application. Using Wi-Fi instead of BLE will allow using older hardware not supporting BLE and/or provide redundancy in the event that Bluetooth has been switched off on the client device.

The Wi-Fi smart beacon will emulate an access point and broadcast an SSID in a special format that will change every 30 seconds as per TOTP specifications. The format will contain a constant system id, location id and a one-time password; having a constant system and location id will allow distinguishing the SSID amongst others, whereas one-time password will be used to authenticate the user and validate a path. By analogy with BLE smart beacons, the location ID will serve as device's UUID.

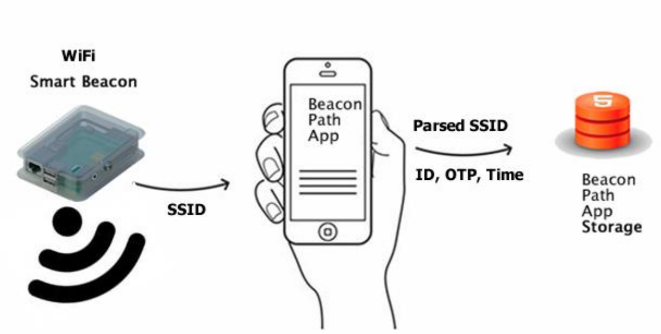


Fig. 4. Wi-Fi Smart Beacon data flow

Implementing SSID broadcast is possible on any commodity hardware including miniature systems like Raspberry PI or even SOHO Wi-Fi routers. The routers running any version of Linux based operating system, such as DD-WRT are relatively easy to be reprogrammed in order to transmit periodically changing SSID broadcasts [19].

Only SSID scan functionality of the mobile device is utilized in this use-case, the mobile application only needs to list down the available SSIDs and there is no need to connect to the Wi-Fi beacon's network. The possibility of searching for other SSIDs broadcasted while connected to one is confirmed and easily reproducible and this means that Wi-Fi smart beacons can be used even while connected to a Wi-Fi network. Once the list of SSIDs have been obtained, the application should search the list for a specific prefix and then parse it to obtain the location ID and the one-time password. E.g. if the format of SSID is "WIFIBEACON_XXX_YYY_ZZZ " , the prefix to look for will be WIFIBEACON, and further on, XXX- is the system ID, YYY- the location ID, and ZZZ is the current one-time password.

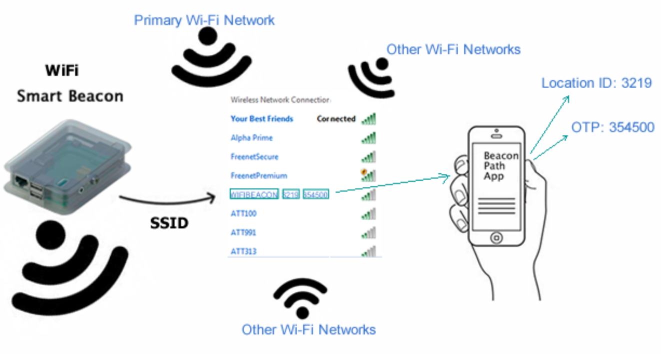


Fig. 5. Wi-Fi Beacon SSID Parsing

3) *Security analysis*

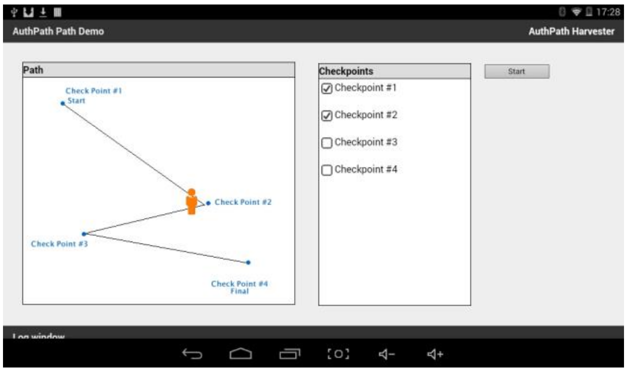
In classic implementation, BLE beacons provide limited level of security, primarily due to the possibility of detecting and cloning beacon IDs. However, the model with "smart" beacons provide higher security in comparison with previous model. Although, the smart beacons are relying on the same BLE technology and attackers can still intercept broadcast packets (the same applies to Wi-Fi beacons, as the SSID is broadcasted in plain-text as well), the risks of "replay" attacks are minimal due to the limited time of validity of the data broadcasted: in majority of TOTP implementations, the period is not more than 30 seconds [5]. In addition, the app should also determine device's GPS coordinates and submit together with harvested broadcast information in order to verify the exact geographic location of the user at the time when the point validation occurred. Additional environmental parameters, such as temperature can be used to avoid wormhole attacks [14]. Additionally to safely identify a device, the parameters transferred to the validation server should be encrypted with device specific key.

IV. BEACON AUTHPATH VALIDATION

In this section, we first detail our model implementation with Estimote devices. Then we discuss what the validation of this prototype has underlined including areas for extensions and current limitations.

A. Prototype Implementation with Estimote

We have developed a prototype path verification app using Estimote Android SDK using Apache Cordova [9]. The app is a combination of an HTML5 interface connecting to android library using Cordova. The object passes nearby (within BLE's allowable distances of 10 cm – 70 meters) certain points equipped with BLE packet emitter or harvester devices. Prototype app has a sample path containing four checkpoints. Two checkpoints were equipped with physical Estimote beacons, 2 remaining checkpoints used virtual beacon apps running on iPhone 5s devices. The interface consists of a graphical path showing the current location of the user and a list of checkpoints. Once a user approaches a checkpoint, the relevant checkbox becomes checked, and the user icon animation of moving to the checkpoint appears. The algorithm checks the order of the beacons ranging, e.g. if checkpoint #3 has been visited before #2, the checkpoint is ignored. For the prototype, any proximity is being considered valid (the TX power value is not checked). The Minor value of beacons are used as their identifications.



☑ Checkpoint #1
☑ Checkpoint #2
☐ Checkpoint #3
☐ Checkpoint #4

Fig. 6. AuthPath application interface

B. Prototype Validation Outcome Discussion

A simple implementation of an Estimote SDK based app has demonstrated how a physical path can be validated using a set of BLE beacons. The exact approach can be used for a real-life application with some improvements: the identification of BLE beacons should be based on UUID with Estimote security enabled, and the validation should be done with integration of Estimote Cloud API into the application. The accuracy of the proximity authentication can be improved by checking the transmission power of checkpoints' BLE beacons (RSSI value), e.g. if we want to validate a checkpoint as passed only if the distance was below 20 meters. Although this prototype is vulnerable to replay attacks as described in 3.1.1, it provides a solid base for creating a more secure implementation, where standard beacons can be replaced with smart devices and the path validation data needs to be sent to the sever at the final checkpoint. It is also important to include GPS coordinates in the array of checkpoints and implement device key encryption before submitting to the validation servers.

While the security level of the presented demo application is disputable due to technology limitation, the app can be used in less strict environments. A possible use area might be Augmented Sports; namely, a smaller scale version of Amateur radio direction finding (also known as radio orienteering) [10] where the only equipment required would be a smartphone and a path of BLE beacons need to be followed showing the best speed and accuracy possible.

An alternative to Apple's iBeacon communication protocol, Eddystone by Google [21] should also be researched in this context. In addition to standard BLE beacon features, Eddystone provides a new type of packet: Eddystone-TLM, which provides telemetry data (TLM stands for "telemetry"). Telemetry data can help to verify the authenticity of a beacon with further accuracy. Different from iBeacon that only operates with numeric IDs, Eddystone can transmit data packets in URL format, which makes it much easier to use for application development.

As already described above, older hardware and mobile operating systems do not fully support Bluetooth 4.0. In addition, many users still perceive Bluetooth as a battery hog. Therefore, for a number of real-life applications BLE might be a less favorable option.

We are proposing the following as alternative solutions to avoid using BLE in the use-cases researched:

1) Near field communication (NFC) [18]– a technology similar to iBeacon working on shorter distances (~ 10 cm)

2) Signals of opportunity (SoOP) [7] based geo-location. This solution uses no beacon devices at all and is more accurate outdoors. However, some companies are promising to provide better accuracy [8]

For Wi-Fi smart beacons, we have done a surface research and it seems to be quite easy to be accomplished, however it is not supported out of the box by Cordova/PhoneGap and a custom plugin will need to be developed [20].

V. CONCLUSION

In this paper, we have proposed a new authentication model based on physical location of a person wearing or passing by BLE beacons. We have validated a part of the overall concept - BLE Beacons based path tracking - with a real implementation and critically evaluated its weak points and proposed potential improvements. Future work will involve larger scale deployment and tests in the shops of a real smart city in the mountain, i.e., a smart ski resort. Creating prototype Smart Beacons, both BLE and Wi-Fi based should be included as a part of further research.

References

[1] F. Stajano, “Security for Ubiquitous Computing.” John Wiley & Sons, 2002

[2] Chen R., Guinness R. Geospatial Computing in Mobile Devices. Artech house, 2014

[3] What is iBeacon? A Guide to Beacons | iBeacon.com Insider [WWW Document], n.d. URL http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/ (accessed 11.6.14)

[4] Estimote Beacons — real world context for your apps [WWW Document], n.d. URL http://estimote.com/ (accessed 11.6,14)

[5] RFC 6238 - TOTP: Time-Based One-Time Password Algorithm [WWW Document], n.d. URL https://tools.ietf.org/html/rfc6238 (accessed 11.6.14)

[6] Bring your own device - Wikipedia, the free encyclopedia [WWW Document], n.d. URL http://en.wikipedia.org/wiki/Bring_your_own_device (accessed 11.6.14)

[7] Chen, R. Ubiquitous positioning and mobile location-based services in smart phones. Hershey, PA: Information Science Reference. 2012

[8] Merlin [WWW Document], n.d. URL http://merlintek.com/ (accessed 11.7.14)

[9] Apache Cordova [WWW Document], n.d. URL http://cordova.apache.org/#about (accessed 11.26.14)

[10] Amateur radio direction finding - Wikipedia, the free encyclopedia [WWW Document], n.d. URL https://en.wikipedia.org/wiki/Amateur_radio_direction_finding (accessed 11.17.14)

[11] What is Secure UUID and how does it work? – Estimote Community Portal [WWW Document], n.d. URL https://community.estimote.com/hc/en-us/articles/204233603-What-is-Secure-UUID-and-how-does-it-work- (accessed 11.26.14)

[12] Bluesmart: World’s First Smart, Connected Carry-On | Indiegogo [WWW Document], n.d. URL https://www.indiegogo.com/projects/bluesmart-world-s-first-smart-connected-carry-on (accessed 11.26.14)

[13] Anind K. Dey. Understanding and Using Context. Personal and Ubiquitous Computing. 2001

[14] YounSun Cho, Lichun Bao, and Michael T. Goodrich. LAAC: A Location-Aware Access Control Protocol. In 2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services. IEEE, 2006

[15] Computer Connector - Two-factor Authentication SAASPASS [WWW Document], n.d. URL https://www.saaspass.com/download/two-factor-authenticat-computer-connector-apple-mac.html (accessed 11.17.14)

[16] van Rijswijk-Deij, Roland . Simple Location-based One-time Passwords - bringing location to the cloud in a secure and simple way. Radboud University Nijmegen. 2013

[17] D. Namiot and M. Sneps-Sneppe “Wi-Fi proximity as a Service”, Think Mind SMART 2012, The First International Conference on Smart Systems, Devices and Technologies. 2012

[18] About the Technology. NFC Forum . [ONLINE] Available at: http://nfc-forum.org/what-is-nfc/about-the-technology/. (accessed 23.02.15)

[19] Heldenbrand, David, and Christopher Carey. "The linux router: an inexpensive alternative to commercial routers in the lab." Journal of Computing Sciences in Colleges 23.1 (2007): 127-133

[20] Cordova - Can Phonegap query the wifi system and return network names? - Stack Overflow. 2015 [ONLINE] Available at: http://stackoverflow.com/questions/4824841/can-phonegap-query-the-wifi-system-and-return-network-names/6769269#6769269. (accessed 23.02.15)

[21] “Beacons | Google Developers”. 2015. [ONLINE] Available at: https://developers.google.com/beacons/?hl=en (accessed 23.08.15)