# Demo Abstract: TagAlong: A Free, Wide-Area Data-Muling Service Built on the AirTag Protocol

Alex Bellon
abellon@ucsd.edu
UC San Diego

Alex Yen
alyen@ucsd.edu
UC San Diego

Pat Pannuto
ppannuto@ucsd.edu
UC San Diego

## ABSTRACT

We demonstrate how to leverage Apple's Find My protocol, most well known as the underlying protocol of the AirTag, for arbitrary data-muling. This provides a new "infrastructure-free" deployment option, where areas with frequent human activity can take advantage of this zero-cost backhaul network. While there are severe limitations (e.g. no acknowledgement channel back to the sending device), Find My-based networking could still be a highly reliable backhaul with sufficient transmission redundancy and knowledge of deployment context.

In this demo, we allow users to send arbitrary data to devices that will forward the data to the Find My network. The data is then recovered from Apple's servers and displayed on a status page. Critically, we will not deploy any of our own intermediate infrastructure and will instead rely on a sufficient density of iPhones and other Apple devices from the demo audience to backhaul data from our demo.

**ACM Reference Format:**
Alex Bellon, Alex Yen, and Pat Pannuto. 2022. Demo Abstract: TagAlong: A Free, Wide-Area Data-Muling Service Built on the AirTag Protocol. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3560905.3568085

## 1 INTRODUCTION

The intended functionality of AirTags is to track items that can be lost (e.g. keys, wallet), though there is ample concern over the actual device usage; while good at tracking inanimate objects, many are concerned with the ability to track and stalk people as well [2, 6, 7, 10, 11]. We exploit additional functionality of AirTags and surface the fact that AirTags—and Apple's underlying Find My protocol—can also be used as data mules. AirTags themselves are Bluetooth Low Energy (BLE) devices that broadcast BLE advertisement packets, which are then picked up by nearby Apple devices that have the "Find My" application installed. It is these nearby Apple devices that add their GPS location and forward along the data to Apple's servers for later access. If the role of the AirTag can be altered, then the functionality of the device can be expanded to carry much more than just location data.

Positive Security [1] first implemented and expanded AirTag functionality that allowed arbitrary data transmission through their "Send My" project. Send My allows ESP32 microcontrollers to act as AirTag-like devices, using surrounding Apple devices to transmit data over the Find My protocol. Users can set the payload to transmit in the firmware, and the payload is transmitted bit-by-bit in BLE advertisement packets over the Find My protocol. In essence, devices with the Send My firmware can transmit data to Apple devices; Apple devices will upload that data to the Find My network. The data can then be downloaded through Positive Security's "DataFetcher" application, which is built on top of OpenHaystack [3, 4]. OpenHaystack retrieves data from devices that upload data to the Find My network based on a customized device ID. All of this work builds upon previous efforts to reverse engineer the Find My protocol and the AirTag's advertisement packet format down to each byte [5, 8, 9].

We expand on this technology to allow for a greater bandwidth of data muling using Apple devices and remote payload entry without the need to reflash the ESP32 firmware. An example diagram of our application is shown in Figure 1. In this setup, we will rely on nearby iPhones from the audience to act as the infrastructure that will ferry our data. In Section 2, we dive further into the implementation

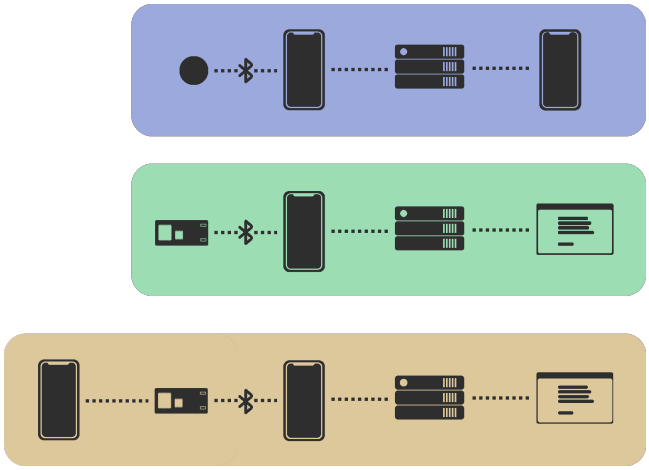

**Figure 1: Comparison of information flow: Find My, Send My, TagAlong** The blue box represents the functionality of plain Find My: the AirTag sends Bluetooth advertisements, which are detected by nearby iPhones, uploaded to Apple's servers, and then can be retrieved by the devices of the AirTag owner. The green box illustrates how Send My works: instead of an AirTag, an ESP32 sends out Bluetooth advertisements, and a specialized application is used to retrieve the data from Apple's servers. Finally, the orange box represents TagAlong, adding in a remote device to set the payload of the ESP32 microcontroller.


*SenSys '22, November 6–9, 2022, Boston, MA, USA*

ACM ISBN 978-1-4503-9886-2/22/11.
https://doi.org/10.1145/3560905.3568085

details and motivation of this demo. We then discuss the specifics of our demo in Section 3.

## 2 IMPLEMENTATION AND MOTIVATION

In this section, we elaborate on the details of our application, and how we built it. The original Send My program flashes the ESP32 microcontroller with firmware that will act similarly to an AirTag, which broadcasts a fixed payload over BLE adversisement packets. Nearby Apple devices (e.g. iPhones) will hear the broadcasted message and forward the data to Apple's Find My network when it has access to the Internet. The data can then be retrieved through the DataFetcher application installed on a macOS device. More details on the installation process and application functionality can be found through Positive Security's Send My GitHub repository [12].

Since Positive Security's Send My and DataFetcher applications only allow devices to send predetermined messages hardcoded into the firmware, we modify the software to enable ESP32 microcontrollers to remotely accept new payloads in addition to increasing the amount of information transmitted in each BLE advertisement packet. Currently, Send My encodes one bit of the payload in each advertisement message, in addition to values denoting the bit and message index. We expand upon the ESP32 firmware to allow for a larger amount of data to be sent in each BLE advertisement packet. In the original firmware, there are nine padding bytes of value `0x00` in each packet and one byte containing the actual bit value that is transmitted by the packet. In our firmware, we construct a new packet format that allows for up to 20 bytes of data to be sent per packet, and condenses metadata fields from 12 bytes to 4 bytes.

Additionally, we further generalize Send My's functionality to change the payload data remotely. Originally, the payload value is only set in the firmware; changing the payload requires changing the firmware and then reflashing the firmware to the ESP32. We add functionality to receive new payloads from devices remotely without any firmware updates. The ESP32 connects to the device through our custom web application to receive new user payload information. It will then relay the payload over BLE using the Find My network, which can then be retrieved using the DataFetcher application.

While the ability to remotely transmit arbitrary data over Apple's Find My network is interesting in its own right, this work demonstrates a system which requires no additional infrastructure for deployment. The devices transmitting the data are completely unaware they are ferrying non-Apple data, enabled by Apple's privacy-focused design of the protocol. This presents an opportunity to deploy IoT networks that previously would have needed to deploy additional costly infrastructure.

## 3 DEMO

In this demo, we use audience members' phones as the infrastructure to send a message over the Find My network using an ESP32 that has the Send My firmware flashed onto it. We have an audience member open up a website application to our TagAlong landing page, which will prompt a message to send. Once they enter the message, it will be sent to our ESP32, which will then exfiltrate the data over the Find My network through BLE advertisement packets. Apple devices in the audience will receive the packets, which will in turn upload them to Apple's servers. Finally, we then show that we can retrieve the message through the DataFetcher application on our MacBook.

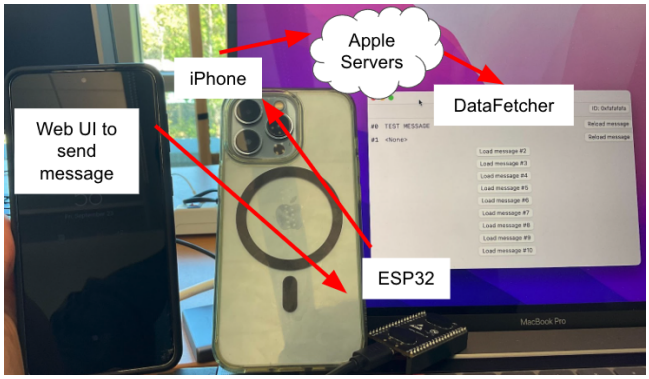


**Figure 2: Photo of experimental setup.** We will show through a website application that we can send data from a mobile phone to an ESP32 with the Send My firmware, which will then upload that data to Apple's cloud servers through Apple devices that are connected to the Find My network. We show that we can then retrieve the data on our Macbook through the DataFetcher application.

## ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2038238. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

[1] F. Bräunlein. Send my: Arbitrary data transmission via apple's find my network, May 2021.
[2] J. Evans. Jealous man used apple airtag to track ex-partner's car. *WalesOnline*.
[3] A. Heinrich and M. Stute. Openhaystack. https://github.com/seemoo-lab/openhaystack, 2021.
[4] A. Heinrich, M. Stute, and M. Hollick. Openhaystack: A framework for tracking personal bluetooth devices via apple's massive find my network. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '21, page 374–376, New York, NY, USA, 2021. Association for Computing Machinery.
[5] A. Heinrich, M. Stute, T. Kornhuber, and M. Hollick. Who can find my devices? security and privacy of apple's crowd-sourced bluetooth location tracking system. *Proceedings on Privacy Enhancing Technologies*, 2021(3):227–245, 2021.
[6] K. Holt. Model brooks nader says someone used an airtag to track her. *Engadget*.
[7] M. Levitt. Airtags are being used to track people and cars. here's what is being done about it. *NPR*.
[8] J. Martin, D. Alpuche, K. Bodeman, L. Brown, E. Fenske, L. Foppe, T. Mayberry, E. Rye, B. Sipes, and S. Teplov. Handoff all your privacy – a review of apple's bluetooth low energy continuity protocol. *Proceedings on Privacy Enhancing Technologies*, 2019(4):34–53, 2019.
[9] T. Mayberry, E. Fenske, D. Brown, J. Martin, C. Fossaceca, E. C. Rye, S. Teplov, and L. Foppe. Who tracks the trackers? circumventing apple's anti-tracking alerts in the find my network. In *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*, WPES '21, page 181–186, New York, NY, USA, 2021. Association for Computing Machinery.
[10] D. Ovalle. Miami-dade cop accused of stalking ex with airtag in latest case involving apple tracker. *Miami Herald*.
[11] A. Scott. Stalker tracks woman with airtag, police say. *WREG Memphis*.
[12] P. Security. Sendmy. https://github.com/positive-security/send-my, 2021.