2018 Third ACM/IEEE Symposium on Edge Computing

Aggio: A Coupon Safe for Privacy-Preserving Smart Retail Environments

Albert F Harris III and Robin Kravets
University of Illinois at Urbana-Champaign
{aharris,rhk}@illinois.edu

Robin Snader
Athetized Network
robin.snader@athetized.net

***Abstract*—Researchers and industry experts are looking at how to improve a shopper's experience and a store's revenue by leveraging and integrating technologies at the edges of the network, such as Internet-of-Things (IoT) devices, cloud-based systems, and mobile applications. The integration of IoT technology can now be used to improve purchasing incentives through the use of electronic coupons. Research has shown that targeted electronic coupons are the most effective and coupons presented to the shopper when they are near the products capture the most shoppers' dollars. Although it is easy to imagine coupons being broadcast to a shopper's mobile device over a low-power wireless channel, such a solution must be able to advertise many products, target many individual shoppers, and at the same time, provide shoppers with their desired level of privacy.**

To support this type of IoT-enabled shopping experience, we have designed Aggio, an electronic coupon distribution system that enables the distribution of localized, targeted coupons while supporting user privacy and security. Aggio uses cryptographic mechanisms to not only provide security but also to manage shopper groups (*e.g.*, bronze, silver, and gold reward programs) and minimize resource usage, including bandwidth and energy. The novel use of cryptographic management of coupons and groups allows Aggio to reduce bandwidth use, as well as reduce the computing and energy resources needed to process incoming coupons. Through the use of local coupon storage on the shopper's mobile device, the shopper does not need to query the cloud and so does not need to expose all of the details of their shopping decisions. Finally, the use of privacy preserving communication between the shopper's mobile device and the CouponHubs that are distributed throughout the retail environment allows the shopper to expose their location to the store without divulging their location to all other shoppers present in the store.

I. Introduction

Retail environments are the current target for deployment of advanced wireless communication, embedded technologies, and streamlined solutions at the edge of the network. The leveraging and integration of technology, such as Internet-of-Things (IoT) devices and hubs, edge-based and cloud-based systems, and mobile applications, can enhance the shopper's experience and increase the store's revenue. By tracking the shopping path of a user via advanced vision techniques (*e.g.*, Amazon Go [1]) or by leveraging low-power wireless devices (*e.g.*, Google's URI Beacon [2]), prototype systems are aiming to provide useful product information and buying incentives to encourage shoppers to make store-influenced purchases. However, current retail solutions mostly focus on counting products in shopping carts [1] or tracking inventory in the store [3], [4]. Although these are both very important components of managing retail spaces, such advanced technology can also be used to bring shopping incentives and product coupons into the age of IoT and directly to the shopper's mobile device.

While it is clear that shoppers want to save money and stores want to sell more products, innovative retail systems must not sacrifice shopper privacy to achieve retail-related goals. Systems, like Amazon Go, leave all privacy control with the store due to the use of vision-based tracking. Vision-based systems in fact necessarily remove all concepts of privacy as a first step. Simply by walking into an Amazon Go Store, shoppers are automatically identified, tracked, and charged. While this may be desirable to some shoppers, others may want individual control of their information exposure through intelligent use of wireless communication technologies, such as Bluetooth Low Energy (BLE) [5][1]. At any given time, a shopper may want to hide their identity, movements, and/or purchasing information. On the other hand, they may be willing to expose some information in exchange for concrete benefits (*e.g.*, EZpass provides faster and discounted road-toll payments, frequent-shopper supermarket programs give fuel discounts and discounted shopping). The key to convincing shoppers to expose more information is to offer them increasingly enticing incentives, while still maintaining sufficient levels of privacy.

Coupon distribution is a well-known, powerful way to affect the spending habits of shoppers. Financially, the global mobile coupon market is predicted to experience a compound annual growth rate of more than 73% by 2020 [6]. Most estimates claim that 80% of the total number of consumers use some form of coupons [7], [8]. Further studies estimate that 79% of shoppers "would find it useful to download money-off coupons to their phones," and 73% would like to "receive instant money-off coupons as they pass by an item in a store" [9]. Essentially, coupons should be presented to the shopper when they are near the products to be the most effective. However, it is also important to the store that only registered shoppers can receive and use specialized or targeted, reward-based coupons. While physical coupons can be placed in specific locations to facilitate such localized coupon distribution, we

[1]Additionally, by combining multiple technologies (*e.g.*, accelerometers, GPS, BLE), a broader array of information with more granularity can be achieved as compared to vision-only systems. This leads to a broader array of services being offered at a finer-grained level of control.

978-1-5386-9445-9/18/$31.00 ©2018 IEEE
DOI 10.1109/SEC.2018.00020

propose an IoT-based solution for managing the placement and distribution of electronic coupons.

Clearly, an IoT-based solution operating at the edge of the store's ecosystem that can directly transfer coupons to particular shoppers at particular times has the potential to impact retail spaces in an extremely dramatic fashion. However, distribution of the targeted, locally-distributed coupons desired by shoppers comes with privacy and security risks as well as fundamental technical challenges. Given the number of products and shoppers in large retail spaces, bandwidth and energy consumption on any chosen wireless channel quickly become strained. For example, Target Corporation has an average of 1.2 million daily visitors across its 1,834 stores [10] and the average grocery store has over 40,000 products [11].

In this context, we present the design and implementation of Aggio, a BLE-based, IoT ecosystem that integrates IoT devices and hubs with local edge-based solutions to enable a store-wide coupon distribution system designed around a secure CouponSafe. Aggio's CouponSafe, similar to a user's password safe, stores all coupons available to a given shopper. However, the coupons can only be unlocked (*i.e.*, unencrypted) when a shopper receives a coupon message from the retail space that includes the appropriate key. By storing the CouponSafe on the shopper's mobile device, the shopper can perform local lookups for coupons without exposing their interests to the other potentially malicious users in the retail environment. By providing personalized CouponSafes to shoppers and requiring a store-provided coupon key, the retail space can control which shoppers can unlock and use which coupons.

The Aggio IoT ecosystem has four main components: the CouponShopper app, CouponSafes, CouponHubs, and a CouponCloud service. When a shopper enters a store, they register with the CouponCloud, which transfers the personalized CouponSafe for this visit to the CouponShopper. The CouponSafe is populated with coupons for that shopper based on their rewards-program level or shopping history, but also contains coupons that the store can unlock at a given time or in a given location. CouponHubs are IoT hubs located throughout the store to distribute coupon messages that contain a key to unlock a particular coupon in the CouponSafe. Aggio utilizes a novel encryption scheme to encrypt coupons in CouponSafes in such a way that decrypting the entry with the given key in a single message results in different information for each shopper or each group (*i.e.*, different discount amounts). Essentially, a single message allows different groups or shoppers to unlock different levels of related coupons. The same novel use of personalizing coupons enables Aggio to use a single message to unlock coupons for different products in each CouponSafe, providing efficient use of the limited BLE bandwidth. To enable localized coupon distribution in the vicinity of specific products, Aggio users can coordinate their location with the CouponCloud or directly with the CouponHubs. To maintain shopper privacy, each communication link throughout the system utilizes privacy and encryption protocols.

The benefits of Aggio come directly from its design. The novel use of cryptographic management of coupons and groups allows Aggio to reduce bandwidth consumption, as well as reduce the computing and energy resources needed to process incoming coupons. Through the use of the local, protected CouponSafe on the shopper's mobile device, each shopper does not need to query the cloud and so does not need to expose all of the details of their shopping decisions. Finally, the use of privacy preserving communication between each shopper's mobile device and the CouponHubs, allows shoppers to expose their location to the store without divulging their location to everyone in the store. In this paper, we evaluate the design of Aggio through a number of interaction models that each provide a tradeoff between complexity, bandwidth utilization, number of shoppers supported, and energy efficiency.

The rest of this paper is structured as follows. Section II presents the design space that Aggio implements, highlight the tradeoff supported by Aggio in terms of system resources and shopper support. Section III presents the corresponding security and privacy concerns brought about by the design space. Section IV presents the design, implementation, and testing of Aggio, including its components: The CouponCloud (Section IV-A); CouponSafes (Section IV-B); CouponHubs (Section IV-C); and the CouponShopper mobile application (Section IV-E). Finally, Section V presents some conclusions and future directions.

II. Supporting Electronic Coupons

Electronic coupons are a powerful tool for use by stores to entice shoppers to buy new or discounted products. Since 73% of shoppers indicated that they would like to "receive instant money-off coupons as they pass by an item in a store" [9], the most effective system must be able to determine what products are in the proximity of the shopper and must be able to deliver coupons quickly to that shopper.

The challenge comes from getting the coupons to the right shopper at the right time. Finding the correct shopper depends on determining their shopping habits and willingness to share personal information, in addition to the current location of the shopper. Since each shopper may want to expose different information about themselves, stores can group the shoppers into different discount groups. Supporting timely delivery of coupons to shoppers near a given product requires location information about the shopper. Again, a shopper may or may not be willing to share this information. To illustrate these issues, consider the following shoppers' situations: an unregistered shopper may stay hidden; a bronze-level shopper may tell the store their identity but not want to share location information; a silver-level shopper may allow location tracking for the day; and a gold-level shopper may allow tracking all month long. The store can then offer different discounts based on the personal information exposure as well at the shopper's location, and so use electronic coupons for each group and in specific locations within the store.

While this desired system of coupon distribution is rather simple to understand, supporting such electronic coupons leads to numerous challenges, both in terms of resource management

and privacy and security. In this section, we explore the system design space. In the following section, we address parallel security and privacy challenges. In the context of system design, there are five main design axes: Centralized vs. Distributed Localization, Centralized vs. Decentralized Coupon Distribution, General vs. Targeted Coupon Distribution, Local vs. Remote Coupon Storage, and Individual vs. Combined Coupon Transmission. Given the diverse demands of shoppers and stores, Aggio has a configurable design, supporting the entire design space. However, the key contributions come from the mechanisms designed to support privacy-preserving local and targeted distribution, maintaining secure access to locally stored coupons, and our novel cryptographic approach to combined transmission.

A. Centralized vs. Distributed Localization

Although not all shoppers will want to share their location, research shows that coupons are most effective when they can be distributed near the product at the time the shopper is approaching [9]. Electronic coupons are perfectly suited to leverage this fact. To enable location sharing while addressing shoppers' privacy concerns, it is necessary to support different types of localization. In a centralized approach, the shopper provides location information (*e.g.*, in terms of RSSI values of nearby WiFi APs) and a retail store server tracks the shopper through the store. The centralized approach relies on the accuracy of the chosen localization technique [12]–[26], each of which trades off accuracy for complexity, energy consumption, resolution, speed, and/or privacy. In a distributed approach, each shopper's device can localize itself based on beacons from WiFi or BLE devices in the store. The shoppers' devices could also beacon to allow fined-grained localization with nearby BLE devices. If the latter approach is used, these beacons must be secured to prevent other shoppers in the store from being able to track them (see Section III). We do not consider external devices for localization, such as the video surveillance in Amazon Go stores, since they do not provide any control to the shopper.

B. Centralized vs. Decentralized Coupon Distribution

The main role of any coupon distribution system is getting the coupons to the shoppers. To distribute electronic coupons, a message must be sent from the coupon distribution system to a shopper's mobile device. That message can either contain the coupon itself or an identifier that can be used to retrieve the coupon. Transfer can be achieved in a centralized manner using a communication path from a centralized retail store server to a particular shopper's device or locally using one-hop communication from an IoT-enabled device in the store directly to that shopper's device.

The centralized approach simplifies distribution by leaving all coupon management to a centralized retail server and does not require any extra devices in the store for delivery, since all coupons are distributed from the centralized server to each shopper's mobile device through the Internet. The timeliness of coupon delivery is tied to the load on the central server or service and the delay for the coupon to be delivered to a particular shopper over potentially longer routes through the Internet. Centralized distribution does not have direct access to a shopper's location, and so depends on centralized localization or distributed localization that has been sent back to the sever to support localized coupon distribution. The combination of slower localization and longer communication routes has a high chance that the shopper is no longer near the product that is the subject of the coupon, and so limits the effectiveness of such coupons. Additionally, indoor localization is required to generate the necessary granularity (10s of feet) to effectively deliver coupons when a shopper is near the relevant target. This too makes a centralized solution infeasible.

Decentralized distribution relies on a local, low-power communication channel to each shopper's device, significantly reducing delay. Additionally, the shorter communication range of protocols like BLE ensure that only shoppers near the product receive the coupons delivered from distribution devices near those products. If shopper location is not known, coupons can be distributed based on some specified store schedule. However, if shopper location is available, coupon distribution can be triggered by the presence of a shopper. Compared to centralized distribution, decentralized distribution requires more infrastructure throughout the store, potentially populated with limited-energy devices to support easy reconfiguration. However, such devices can be manufactured at very low cost.

C. General vs. Targeted Coupon Distribution

Not all coupons are intended to be used by all shoppers. In general, coupons can be offered to the general public or targeted at specific shoppers or groups/categories of shoppers. Physical general coupons are commonly found in newspapers or fliers in the entry halls of retail spaces. Essentially, general electronic coupons are delivered to and can be used by anyone that happens to be on a distribution list or have electronic possession of the coupon. General coupons are fully transferable (shopper A can give their copy to shopper B for use), and are unrelated to either the shopper's current location, or their previous purchasing habits. However, marketing research consistently shows that directed, electronic couponing is the most effect way to improve sales [7]–[9]. In response, targeted coupons focus on giving deals to particular shoppers based on shopper classifications (*e.g.*, gold loyalty member), prior shopping habits (*e.g.*, buys milk and cereal frequently), and/or the shopper's current location in the retail space.

Membership-based or personalized targeted coupons can be used to reward a shopper for their loyalty or for providing information about their demographics and shopping history. Such coupons can be directed to a particular shopper's mobile device as determined by the store's policies. On the other hand, the introduction of IoT into the retail space enables location-based targeted coupons. Given knowledge of a shopper's location, coupons for nearby products can be sent to the shopper's mobile device, enticing the shopper to try new products or discounting products related to those near the shopper. For

example, if a shopper is located near breakfast cereals, a coupon could be generated for milk, to drive the shopper's path through the store towards the dairy section. Although membership-based targeted coupons do not require location information, location-based targeted coupons obviously do. Location-based targeted coupons can be supported with centralized localization and coupon distribution. However, there is a natural fit with decentralized approaches.

D. Local vs. Remote Coupon Storage

With the enormous number of products and shoppers in a normal retail store at any given time, many coupons may be available, but any particular shopper may only be interested in a fraction of those available coupons. Additionally, to learn about the potential savings from an electronic coupon in a locally delivered message, the shopper's mobile device must do some type of look up.

With remote storage, the shopper's mobile device must either be directly sent the entire set of coupons by the coupon distribution system or do one remote lookup on the server for every message it receives giving it access to a new coupon. This is not only resource intensive, but also always exposes information to the server about the shopper's location and potentially about their product interests, speed through the store, and numerous other privacy-leaking metrics.

Local coupon storage requires that the shopper's mobile device have a database of every coupon that shopper may potentially receive in the store. This database must obviously be protected so that the shopper's mobile device can only access the coupons for which the retail space sends messages granting access. If this database has open access, targeted coupons cannot be supported. To enable local coupon storage while maintaining access control to the coupon database, Aggio, sets up a CouponSafe and delivers it to a shopper when they enter the store. This delivery could be configured to happen via a push from the cloud every morning, or on some other event. For the initial system, we chose on the shopper entering the store so that the shopper would have the freshest possible coupon set. The shopper can only access an entry in the list once local keys are distributed based on proximity to Aggio's CouponHubs (see Section IV-C).

E. Individual vs. Combined Coupon Transmission

Electronic coupon distribution in large retail stores has the additional challenge of supporting a large number of products and delivering a large number of coupons to diverse shoppers. Targeted coupons for every product and for every shopper or group results in an explosion in the number of coupons to be sent. Low-power IoT-based devices using, for example, BLE, rapidly have their resources become completely overloaded [3]. To compensate for this, Aggio takes a novel cryptographic approach and enables the combination of multiple coupons into a single message. By using careful cryptographic encoding, Aggio's CouponHub can send a single message that translates to multiple different coupons depending on the key or keys the shopper has been given for decrypting these messages (see Section IV-F).

III. Privacy and Security for Smart Retail

Any public retail system must safeguard the security and privacy of the shoppers being served. Current IoT-based retail solutions suffer from a general lack of privacy and security protections [5], [27]–[31]. Some early work focused on attempting to allow users to participate in loyalty programs while hiding their true identity from the retailer [32], [33], but generally had no concern for third-party attacks or privacy leaks beyond information directly shared in old systems (*e.g.*, shopper name and phone number). In fact, mobile applications that interact with their environment and IoT-enabled devices have been found to be devoid of such protections regardless of the intended deployment environment [30]. Supporting electronic coupon distribution at a system level necessarily involves the transfer and tracking of various levels of identifying information. This section focuses on the specific privacy and security leak points in the electronic coupon system space.

A. Shopper Information to the Coupon System

To support electronic couponing, shoppers in the retail store can choose to send varying levels of personal and/or tracking information to the system. Any time information flows outward from a shopper's mobile device, malicious devices within reception range have the opportunity to break the shopper's privacy. There are two general categories of concern when a shopper's mobile device needs to share information with the system: indirect privacy breaches and direct privacy breaches.

1) Shopper Information: Direct Privacy Breach: Whenever personal information is sent, there is always the possibility of a direct privacy breach in which the information is intercepted. To prevent such direct privacy breaches, Aggio utilizes an AES-based encryption protocol designed to be energy-efficient, support different levels of exposed shopper information, and be easily configurable each time a shopper enters the retail space, without the need for the shopper to remember information from a previous shopping trip.

2) Shopper Information: Indirect Privacy Breach: Although they do not contain any shopper specific information, protocols like BLE and WiFi have certain characteristics that can be used to identify and track a user that are unrelated to the actual data sent, resulting in an indirect privacy breach. For example, BLE, when used in its standard advertising mode, transmits a beacon periodically that contains various fields that do not typically change value (*e.g.*, the MAC address in the iBeacon format [34]). These fields can be utilized to identify and track a shopper as they move through the retail environment. Other fields do change, but can still lead to privacy leaks (*e.g.*, RSSI values). Such information can be used to track a shopper's location throughout a retail environment [28]–[30].

B. Coupon Messages to Shopper Mobile Devices

Instead of directly transmitting a coupon to a shopper, the retail system can send a message identifying a coupon that

the shopper's mobile device can then retrieve from the retail server. However, this message must be protected so that only the intended shopper can use the message to retrieve and utilize the coupon.

C. Coupon Requests

When a shopper's mobile device retrieves a coupon that is stored remotely (*e.g.*, as described in Section II-D), malicious eavesdroppers could use the transmission of these coupon-retrieval messages to track a shopper through a store and learn about their shopping habits. For example, if a particular shopper frequently issues coupon retrieval messages when in front of milk, ice cream, and vitamins, an eavesdropper could infer that these products are of interest to that particular shopper. In this way, shopper "fingerprints" could be built by mere observation of coupon-retrieval traffic patterns.

D. Locally-Stored Coupon Database

To support storage of the coupon database on a shopper's mobile device (as in Section II-D), the local coupon database must be secured against the shopper accessing all coupons all of the time. Essentially, the shopper should only be able to access and utilize those coupons for which they have received a message to access. The remaining coupons in the database should remain unaccessible.

IV. Aggio: Electronic Coupon Distribution System

The Aggio electronic coupon distribution system supports the range of design decisions discussed in Section II. The system architecture is focused on providing the services needed given the resource constraints in terms of bandwidth and energy while maintaining shopper privacy through the use of novel algorithms. These algorithms are used to maintain the security of the coupons, target coupons at specific shoppers, and maintain shopper privacy. Aggio is composed of four main components: a cloud service (called the CouponCloud), multiple IoT-enabled in-store devices (called CouponHubs), a coupon storage component (called a CouponSafe), and a shopper mobile application (called CouponShopper). While the algorithms presented in this section are all described in the context of electronic coupons, our algorithms and system components are applicable to other IoT environments and applications.

Figure 1 depicts the Aggio system components along with their main communication pathways. A single Aggio deployment consists of one CouponCloud and large numbers of the supporting components. The following subsections describes each of these components and their interactions in detail. To illustrate the system in a coherent manner, the remainder of this section follows a shopper as they first enter an Aggio-equipped retail environment.

A. CouponShopper: Beginning the Experience

Each Aggio-enabled retail environment sets up an instance of a CouponCloud, which is implemented as a Google App Engine Cloud service [35] using the Go programming language [36]. The CouponCloud has three fundamental components: a mechanism to access CouponSafes, a client key distribution component, and a client seeding component. Shoppers utilize the CouponShopper mobile application as the user-interface to Aggio-enabled retail environments (see Figure 2).

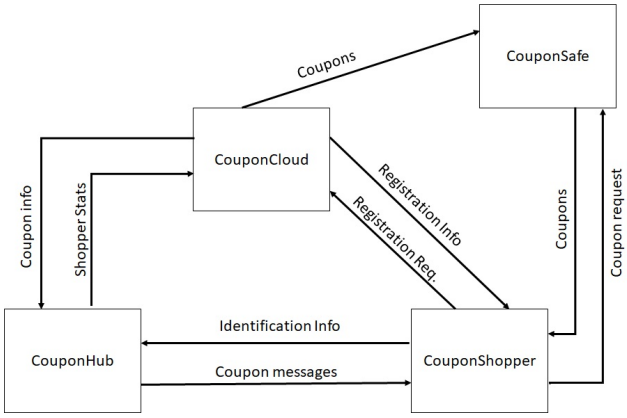

Fig. 1. Aggio: System Architecture

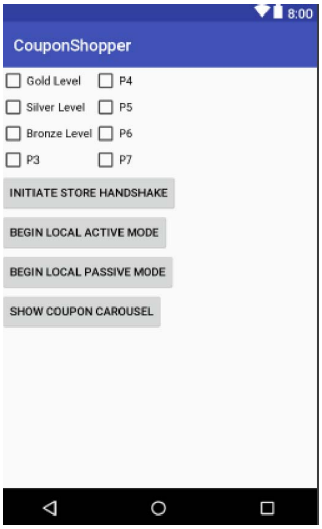

Fig. 2. The CouponShopper Mobile Application

Prior to installing CouponShopper, a shopper registers for a rewards account with the Aggio system. This registration is similar to current shopper rewards registrations except that Aggio users can select what level of information to share with the retail environment. For example, shoppers can choose to share their actual identity for use in tracking at every visit, a one-time use identity for tracking a shopper only during each visit, or no identity sharing at all [5]. Furthermore, shoppers can choose to share various information about their purchasing habits and desires. Finally, the retail organization itself can choose to classify shoppers based on any metric they choose. Table I depicts the classification matrix used by

Shopper Classifications	
Identity Tracking	Real/Permanent ID Temporary/One-time ID No ID
Preference Sharing	Cart Contents Favorite Categories Lemonade
Reward Classification	Bronze Level Silver Level Gold Level

TABLE I
TRACKED SHOPPER CLASSIFICATIONS

Aggio for the tests and experiments presented in this paper. Aggio creates these classifications by combining cryptographic keys. Therefore, the category space is essentially limitless and easily extensible to fit whatever business model is desired by any particular retail environment.

Once the shopper is registered, the CouponShopper application is downloaded and installed on the shopper's mobile device. When a shopper first enters a retail environment, the CouponShopper application on their mobile device automatically performs a handshake with the store's CouponCloud. The CouponShopper handshake interacts with the client seeding component and the client key distribution component. Figure 3 shows the CouponShopper to CouponCloud handshake. The prototype of CouponShopper used for the tests and experiments presented in this paper does not automatically register with the CouponCloud. Instead, there is a button (see Figure 2) that initiates this handshake. To facilitate automatic handshakes, CouponShopper could use GPS or WiFi localization and geo-fencing to determine when the shopper has entered or left the store. The use of these techniques is well-known and beyond the scope of this paper.

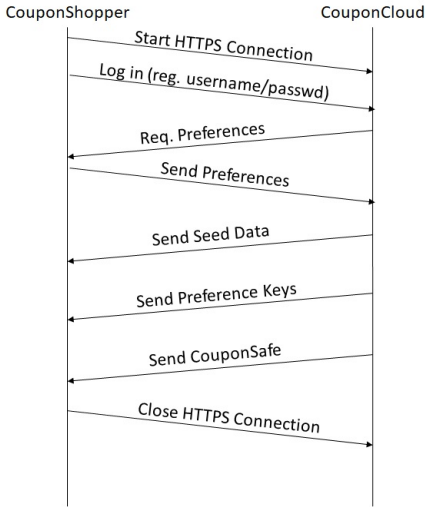

Fig. 3. The CouponShopper Handshake

To initiate the handshake upon entering the retail environment, CouponShopper sets up an HTTPS connection with the CouponCloud. Once the secure connection is established, CouponShopper authenticates the shopper with the CouponCloud using the credentials established during the online registration. The CouponCloud sets up a preferences context for the shopper's current shopping visit. If the shopper established a willingness to share further information for the visit (*e.g.*, current product interests for the trip), the CouponCloud requests the information, called `local_preferences`. The CouponShopper returns the `local_preferences` to be used for the current shopping trip. Finally, and potentially based on the `local_preferences`, the CouponCloud securely transmits up to three components for use during the shopping trip.

1) **Seeds**: If CouponShopper is configured for active shopping mode (see Section IV-E1), CouponCloud transmits the seeds needed for active communication with the CouponHubs. The seeds used for the implementation described in this paper are 16,B random numbers. They are used for the initialization of the security and privacy mechanisms used to protect local communication between the CouponShopper and CouponHub (see Section IV-E1).
2) **Preference Keys**: CouponCloud transmits the preferences key chain used for receiving and utilizing coupon messages targeted to the shopper. The preference keys used for the implementation described in this paper are 32,B.
3) **CouponSafe**: If CouponShopper is configured for local CouponSafe storage (see Section II-D), CouponCloud transmits the CouponSafe specifically built for the shopper.

Once the transfer of components is complete, CouponShopper tears down the HTTPS connection to the CouponCloud. Assuming CouponShopper is configured to utilize a local CouponSafe, no further cloud communication is necessary for the remainder of the shopping visit. Although it is important to optimize the handshake and the configuration and transfer of the CouponSafe, this only happens once as the user enters the store, and so typically would have enough time to complete before the user starts shopping.

B. CouponSafe: A Secure Store of Coupons

The CouponCloud is responsible for configuring the shopper's CouponSafe and the CouponHubs throughout the retail environment. Once a shopper is registered with Aggio and is entering the store, a CouponSafe must be configured with coupons to be delivered to the shopper. Additionally, CouponHubs throughout the store may require additional configuration to support the new shopper. The remainder of this section describes CouponSafe and CouponHub configurations.

In the CouponSafe, Aggio stores encrypted coupons for potential future use. The Aggio prototype uses full-color coupons with QR codes (which are the standard to be scanned at the register) containing a GS1-standard complaint coupon code (see Figure 4). The GS1 standard defines the format for generating unique coupon identifications. The code begins with the coupon application code (8110), followed by a 0 and

Fig. 4. Aggio Coupon

the GS1 Company Prefix assigned by the GS1 organization (6-digits). This prefix uniquely identifies a given company. These digits are followed by a 6-digit offer code locally generated. Next is a 1-digit length followed by a save value of that length and another 1-digit length followed by a purchase requirement of that length. These are followed by a 1-digit requirement code indicating a type of purchase requirement (*e.g.*, number of units, value of total transaction, etc.). Finally a 3-digit product family code ends the coupon code. The maximum length for the save value and purchase requirement values are 5 digits each, therefore, the total length of the coupon code is between 25 and 33 digits. [37]. It should be noted that Aggio is sufficiently general to support other types of coupons, for example, coupons that could be used at a register via NFC.

To support localized, targeted, timely coupon delivery, the CouponSafe must prevent shoppers from accessing coupons prior to receiving access messages (called coupon messages). In Aggio, CouponSafes can be co-located either with a shopper's CouponShopper or with the CouponCloud (see Section II-D). Each shopper's CouponSafe is separately configurable. It is important to note that for shoppers choosing to store their CouponSafe in the CouponCloud, each coupon message received triggers a connection to the CouponCloud, potentially leaking privacy information. For example, even if the communication to the CouponCloud is secured using HTTPS, if CouponShopper initiates CouponCloud communication every time a targeted coupon message is received, a malicious snooper can infer product interests based only on where in the store the CouponShopper initiates these communication patterns.

Text and a QR code take very little storage. The CouponSafe can store approximately 200 coupons for every 1 MB on disk. The full-color, text and QR code coupons generated for the prototype presented in this paper (*e.g.*, see Figure 4) are approximately 5 KB on disk. The CouponSafe is implemented as a table with the format shown in Table II. The CouponSafe encrypts the coupons and the coupon GS1 codes, which can be used to electronically redeem coupons and therefore must also be protected, using a shared secret encryption scheme that utilizes AES CTR mode as an encryption primitive [38].

To build the CouponSafe, CouponCloud first builds a list of coupons that could potentially be directed to a shopper. This list is built based upon the user preferences determined during shopper registration as well as any shopper classifications that might apply (*e.g.*, Silver-level rewards member). Once the set of coupons is determined, the CouponCloud uses the assigned cryptographic seeds and keys to build the encrypted CouponSafe according to Algorithm 5. This algorithm assigns an AES initialization vector built from the combination of seed material and a random value. A mapping is built between initialization vectors, assigned keys, and coupon identifiers. This mapping is later used to generate coupon messages to be sent to the CouponShopper to allow access to particular coupons in the CouponSafe (see Section IV-C).

ENCRYPT COUPONSAFE()
1 **while** $(moreCoupons)$
2 **do**
3 $AESInit \leftarrow shopperSeed|random(nextCouponID)\%2^6$
4 $shopperMessages[currentMessage].couponID$
5 $\leftarrow nextCouponID$
6 $shopperMessages[currentMessage].AESInit \leftarrow AESInit$
7 $shopperMessages[currentMessage].key$
8 $\leftarrow shopperKeys[nextKey]$
9 $keystream \leftarrow AES_key(AESInit)$
10 $GS1Encrypt \leftarrow GS1 \oplus keystream$
11 $shopperMessages[currentMessage].key$
12 $\leftarrow shopperKeys[nextKey]$
13 $keystream \leftarrow AES_key(AESInit)$
14 $couponEncrypt \leftarrow coupon \oplus keystream$

Fig. 5. Build CouponSafe

Once the CouponSafe is built, it is transmitted to the CouponShopper (if local storage is chosen). Since each separate coupon is encrypted with a separate key and initialization vector, the loss or breaking of any single key only gains access to a single coupon. The integrity of the entire CouponSafe cannot be compromised by any single broken key. If the CouponCloud is broken, entire CouponSafes could be compromised. However, cloud security is expected to be more carefully protected than any particular mobile device's security. In the next section, we describe the CouponHubs and their role in distributing coupons.

C. CouponHub: In-Store Coupon Messaging

CouponHubs are IoT-enabled devices designed to support decentralized coupon distribution. CouponHubs are designed to be energy-efficient, support shopper privacy, protect the coupons that are distributed, and be remotely updated. To support decentralized coupon distribution, CouponShoppers are equipped with an IoT-supporting radio. The Aggio prototype uses the nRF52832 from Nordic Semiconductor [39] for its BLE [40] support and its rich, low-power capabilities.

CouponSafe		
Coupon Index (6,B)	**Encrypted GS1 Code (15,B)**	**Encrypted Image (5,KB)**
561598	25a6b6cd587541a	2133541...
566518	22b15fddac218d1	8513044...
	...	
568521	a2d5cb654afd225	5210385...

TABLE II
COUPONSAFE: TABLE FORMAT

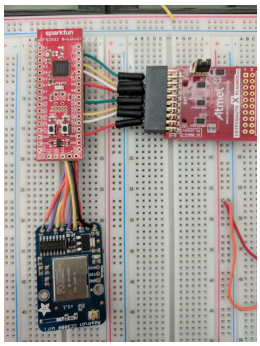

Fig. 6. The CouponHub Breakout Components

While the nRF51822 seems to be more popular, the DCDC converters on the chip are defective, making power control essentially nonfunctional. The 52x chips do not have this defect. The nRF52832 provides all the needed communication functionality between CouponShoppers and the CouponHub. For communication between the CouponHub and the CouponCloud, the prototype uses a CC3000 WiFi chip [41], which supports HTTPS. Finally, to support efficient processing and native AES encryption, the CouponHub prototype uses an ATMEL ATAES132 chip [42]. The ATMEL ATAES132 provides hardware AES primitives that support Aggio's algorithms that require encryption. The initial prototype was tested and designed using various breakout boards (see Figure 6). Using the results of these tests, we designed an integrated CouponHub prototype, which also includes a built-in antennae and an onboard EEPROM for extra memory (see Figure 7).

1) Coupon Messaging: CouponHubs are responsible for delivering messages that allow CouponShoppers to retrieve coupons from their CouponSafe, determine the coupon's value and redeem them. Figure 8 depicts the packet format used to transmit coupon messages from the CouponHub. Coupon messages are transmitted using the nRF52832 BLE radio in passive beacon mode. With an advertising rate of 200 ms, each CouponHub can transmit approximately five coupon messages per second. The average person takes 10 to 15 seconds to walk a store aisle, assuming they are not looking for something. We found experimentally that the BLE signals did not propagate well between aisles and a single CouponHub placed near the center of an aisle could cover the entire aisle. Therefore, each CouponHub has enough time to send 50 to 75 messages during the time a shopper is in a single aisle. In this section, we describe the process by which a CouponHub builds and transmits coupon messages, the messages that contain the information the CouponShopper requires to access a coupon.

Each coupon message packet contains a store identifier to help disambiguate messages between adjacent retail spaces. Each coupon is related to a particular set of preference keys that a shopper must have to access and utilize the coupon. CouponHubs use these preference keys to protect the AES material needed to access the coupon in the CouponSafe. CouponHubs use a modification of AES CTR mode to transmit coupon messages in such a way so that only the intended shoppers can successfully decrypt the AES material used to access the coupon in the CouponSafe. Algorithm 9 is used by the CouponHubs to generate coupon messages.

Formally, a coupon C is made available to shoppers based on any Boolean formula of the form $c(p_1, p_2, \ldots, p_n)$ that can be expressed in negation-free disjunctive normal form, where p_i correspond to the different privilege keys. In particular, if

$$c(p_1, p_2, \ldots, p_n) = \bigvee_{i=1}^{m} c_i(p_i, p_2, \ldots, p_n)$$

where

$$c_i = \bigwedge_{p_j \in C_i} p_j \; \forall i, 1 \leq i \leq m$$

(where C_i represents the set of terms present in the i-th disjunctive clause, then (letting k_i be the privilege key corresponding to privilege p_i) a CouponHub can transmit

$$E_{K_1}(C), E_{K_2}(C), \ldots, E_{K_m}(C)$$

where

$$K_i = \bigotimes_{p_j \in C_i} k_j \; \forall i, 1 \leq i \leq m.$$

In practice, to give a shopper access to a coupon C, a CouponHub first populates the Privilege Key Bitfield based on a particular C_i, then computes K_i as described above. Aggio uses AES in CTR mode to provide a pseudo-stream cipher that is resilient to packet loss without the weaknesses inherent in ECB modes. Thus, the current nonce is concatenated with the current value of the AES counter and this value is encrypted with K_i. The resulting keystream can then be XORed with the message data to be protected corresponding to C to obtain $E_{K_i}(C)$. This process can then be repeated over all values of i (*i.e.*, for all clauses of c).

It is important to notice that the same privilege keys can be distributed to multiple CouponShoppers. In other words, there may be one privilege key for all gold-level shoppers. Therefore, any coupon message sent by a CouponHub and targeted at gold-level shoppers could be utilized by anyone

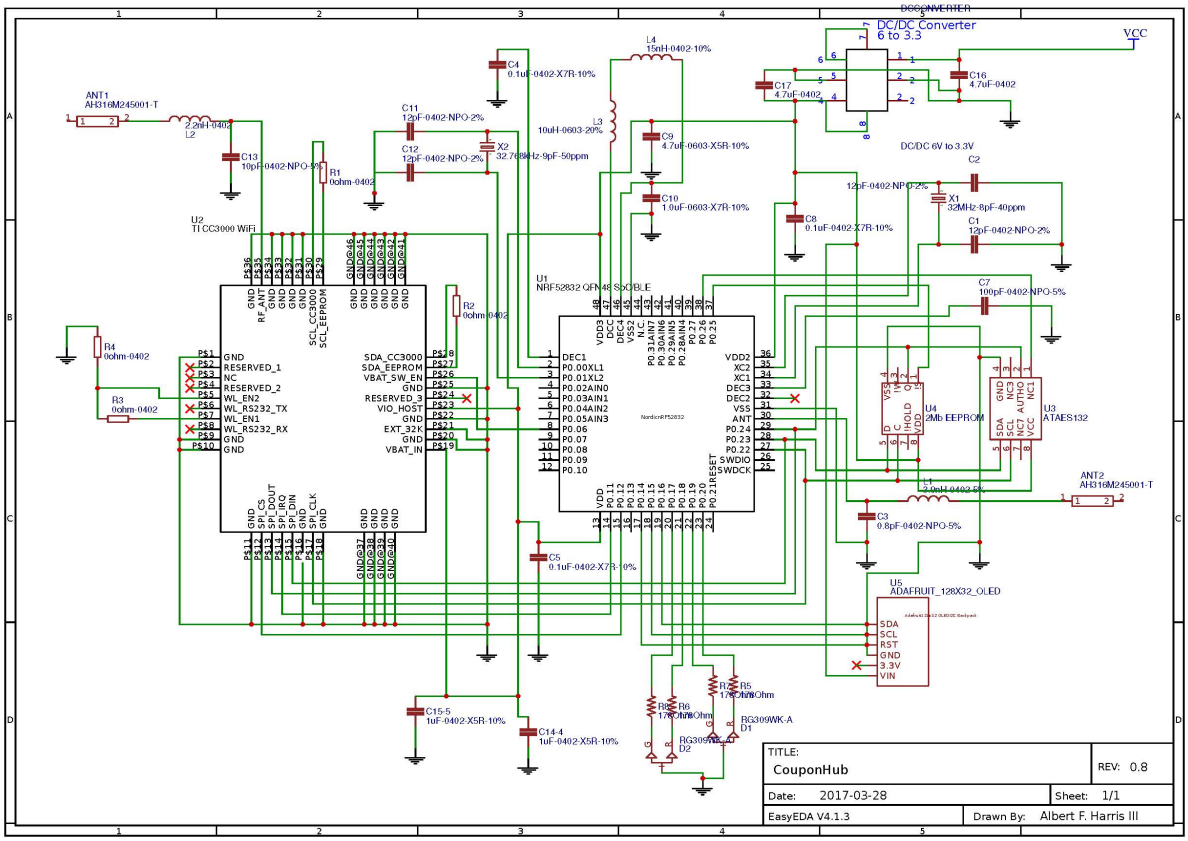

Fig. 7. The CouponHub Architecture

PK bitfield (1 byte)	Store ID (2 byte)	Coupon ID (6 byte)	Nonce (4 byte)	CTR (4 byte)	Encrypted AES Material (14 byte)

Fig. 8. Coupon Message Packet Format

with the gold-level shopper privilege key. Therefore, a single coupon message generated by a CouponHub may be used by numerous CouponShoppers.

Each CouponHub is transmitting coupon messages at an expected rate of five to ten a second, with each message requiring an AES encryption step. Since the CouponHub is performing a large number of encryptions, we wanted to understand the energy impact of utilizing dedicated hardware for encryption as opposed to a software implementation. To explore this space, we modified an AES library commonly used on the Nordic Semiconductor devices, the Wolf (CyaSSL) library [43]. We measured the energy consumption using a BK Precision 2530B digital oscilloscope and a shunt circuit. Figure 10 shows the energy consumption measured in mA and then normalized to the energy consumption of the nRF52832 while transmitting. Both hardware and software encryption are cheaper than transmission, as expected. Given the simple functions required to perform AES encryption, there is a very clear benefit to using specialized hardware. While this may not make a difference on the CouponShopper side, since the number of encryptions and decryptions are not that large (on the order of the number of received coupons), CouponHubs do almost nothing without performing AES functions. These results further validate our decision to add specialized encryption hardware to the CouponHubs.

2) CouponHub to CouponCloud Communication: The CouponHub uses the WiFi radio to connect to the CouponCloud for two primary purposes: to receive the coupon and privilege key information; and to transmit statistics back to the CouponCloud. As described int the previous section, CouponHubs generate coupon messages. The set of messages each CouponHub is responsible for delivering is determined based on a number of factors, at least including the types of products located in the aisle that the CouponHub is servicing. For the testing and experiments in this paper, we assume that the coupon message provisioning occurs offline. The CouponCloud maintains a database of coupons and CouponHub identifications. The CouponCloud then uses this database

```
COUPON MESSAGE()
 1  AESInit
 2        ← shopperMessages[currentMessage].AESInit
 3  AESkey ← shopperMessages[currentMessage].key
 4  couponID
 5        ← shopperMessages[currentMessage].couponID
 6  for j ← 1 to N
 7  do
 8     if usesKey(j)
 9        then bitfield ← bitfield ∨ 2^j
10             K = K ⊕ k_j
11  appendToMessage(bitfield)
12  appendToMessage(shopID)
13  appendToMessage(couponID)
14  appendToMessage(nonce)
15  appendToMessage(counter)
16  key ← AES(nonce||counter, K)
17  encryptedAESMatter ← (AESInit|key) ⊕ key
18  appendToBeacon(encryptedAESMaterial)
```

Fig. 9. Build Coupon Message

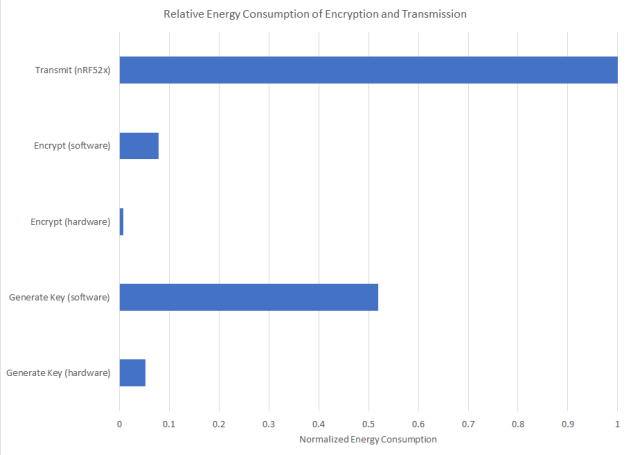

Fig. 10. Encryption Energy Consumption

to deliver the necessary information (AES initialization material as described in Section IV-B) and the privilege key mappings (also determined offline). The CouponHub hardware has sufficient storage that it only needs to download new information from the CouponCloud once a day.

The CouponHub also maintains statistics related to which coupon messages were generated and when. Additionally, the CouponHub maintains statistics as to which, if any, CouponShopper applications were heard in the area. These statistics are stored on the local EEPROM, encrypted for security by one of the hardware encryption routines provided by the ATMEL AES132. These statistics can be transmitted to the CouponCloud periodically as well.

D. Passive Coupon Distribution

CouponHubs can trigger coupon messages either passively or actively. Passive triggering is essentially schedule based. Each CouponHub has a configurable messaging interval (*e.g.*, our CouponHubs used a 200 ms interval). At each message interval, the CouponHub transmits the next coupon message. The ordering of these messages can be determined based on a simple round-robin algorithm, wherein each coupon message gets created and transmitted in order, and no one messages is repeated until every message is transmitted. Alternatively, some pre-determined priority scheme could be used. For example, each gold-level customer message could be transmitted at twice the frequency as the other levels. The common feature among these passive schemes is that there is no guarantee that any shopper will be within range of the coupon messages transmitted. Given that CouponHubs are expected to be battery powered, transmitting messages when no CouponShopper is around is clearly undesirable. However, the system should still support CouponShoppers that do not wish to announce their presence. Therefore, CouponHubs support variable interval messaging. When a CouponHub hears no actively messaging CouponShoppers, a longer message interval is used than when active CouponShoppers are present. Active CouponShoppers are described in the next section.

E. The CouponShopper Mobile App

This section describes how a shopper uses CouponShopper to receive targeted shopping incentives. The CouponShopper application can move around the retail space either in passive or active mode. However, any time a shopper's mobile device transmits information, there is the potential for that shopper's privacy to be disrupted. CouponShopper uses the mobile device's BLE radio for communication with CouponHubs, for the purpose of transmitting simple identifying information that could be used by the CouponHub to provide more targeted coupons to the shopper. There is fundamentally a tradeoff occurring, wherein the retail space offers shoppers more enticing incentives in exchange for sharing extra information. However, while a shopper may be wiling to agree to share extra information (*e.g.*, a one-time identifier that could be used to track a shopper through the store during the current visit), that shopper will not likely want to share such information with other strangers in the store: strangers who are not offering any incentives for that granularity of information. The Aggio BLE protocol uses a self-synchronizing stream encryption algorithm to plug privacy leaks in the standard BLE protocol as well as encrypt the small packet sizes available to BLE.

1) Active Shopping Messages: As part of the handshake described in Section IV-A, a number of seeds are exchanged between the CouponCloud and the CouponShopper. These seeds are used to construct a CouponShopper ActiveModeKey. CouponShopper periodically transmits a packet containing an encrypted version of a shopper identifier that can be used by CouponHubs and the CouponCloud to identify the shopper. This identification depends on the level of information sharing the shopper agrees to. For example, this identification could

be linked to the shopper's real name and address, or it could merely be a temporary identification used to track the shopper during a single retail experience. Algorithm 11 is used by CouponShopper to generate the periodic transmission. It should be noted that the CouponShopper sets its MAC address to a pseudorandom number to avoid being tracked throughout the store. Additionally, because AES CTR mode is being used, even though the shopper identifier is not changing, the encrypted version of the shopper identifier will be different for every transmission.

ACTIVEANNOUNCE()
1 $CTR \leftarrow random(index)\%2^40$
2 $encrShopperID$
3 $\leftarrow shopperID \oplus \text{AES256}_{ActiveModeKey}(CTR)$
4 $setPacketMAC(\text{SHA256}(nonce|CTR))$

Fig. 11. Packet encryption algorithm

2) Processing Coupon Messages: The primary function of the CouponShopper is to process coupon messages and display coupons for the shopper. Upon receiving a coupon message, the CouponShopper must determine if it can decode the message. Wasting energy by attempting to run cryptographic algorithms unsuccessfully is undesirable, so first the CouponShopper checks the preference keys bit field in the coupon message. If the required preference keys match the set kept by the CouponShopper instance, CouponShopper uses the nonce and CTR values from the packet to decode the encrypted AES material that is the payload of the coupon message (see Algorithm 12).

DECODEMESSAGE()
1 **if** $privs \vee \neg bitfield$
2 **then for** $j \leftarrow 1$ **to** N
3 **do**
4 **if** *bitfield* $\wedge\ 2^j$
5 **then** $K = K \oplus k_j$
6 $key \leftarrow AES(nonce||counter, K)$
7 $AESMaterial$
8 $\leftarrow encryptedAESMaterial \oplus key$

Fig. 12. Message Decoding Algorithm

Once the AES Material is recovered, the CouponShopper does a table lookup in the CouponSafe using the 6,B coupon identifier from the coupon message. The encrypted GS1 code and coupon are both decrypted using the AES material recovered from the coupon message, the coupon is popped up on the shoppers mobile device to alert the shopper that a coupon has been received, and the coupon is added to a local, decrypted coupon carousel for later use.

One metric by which Aggio can be tested is the effective delivery rate of coupons given a shopper walking through an aisle. To test this, we deployed a CouponHub in a local grocery store. We deployed the CouponHub in the canned fruits and vegatables aisle, placed on a middle shelf, roughly waist high, near the front of the shelf (*i.e.*, not behind or under any products). The aisle is approximately 18 meters long. The CouponHub was set to transmit a coupon message every 200,ms. To collect data, we arbitrarily selected a product known to be in the aisle, but whose location was unknown to the walker and had the person walk the cart into the aisle starting at one end, locate the item, and then walk the cart out of the aisle at the other end. messages were collected by the CouponShopper application from the time the walk began to the time the walk ended at the opposite end of the aisle. We made numerous such runs. On average, nearly 400,ms passes between reception of coupon messages. Figure 13 shows a representative segment from one such run. BLE is well known to exhibit poor reception rates as interference and contention increases. It is therefore not surprising that, even given a 200,ms message interval, there were times when over half a second would go by without a coupon being received.

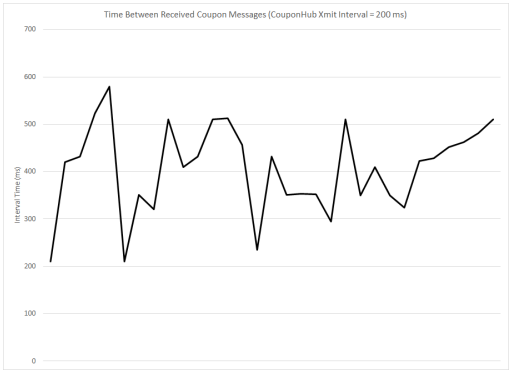

Fig. 13. Time Between Coupon Message Receptions

F. Coupon Message Optimization

One potential performance bottleneck for Aggio is the CouponHub. Given the large number of products in modern retail environments and the large numbers of potential shoppers, CouponHubs covering a single aisle may rapidly run out of resources to serve all necessary coupons. For example, CouponHubs transmitting at coupon message every 200 ms can only transmit about 50 messages in the time an average shopper walks down an average aisle. While this may seem like many coupons, consider the fact that in that aisle, over 100 linear feet of shelving exists, possibly containing over 100 unique products. Additionally, there may be multiple coupons for each single item (*e.g.*, 10% off milk for bronze rewards customers, 15% off milk for sliver rewards customers, and 20% off milk for gold rewards customers). It is not hard to see that bandwidth rapidly becomes overloaded.

However, in this same example scenario, the entire coupon is not in fact different for each level. Instead, they are all related. Aggio is designed to leverage this similarity and allow the CouponHub to transmit a single encrypted message in such a way that, depending on the key used to decrypt the

coupon message, different, predetermined AES material sets are derived. The AES material sets can then be utilized to access different versions of the coupons: one for each level (see Figure 14).

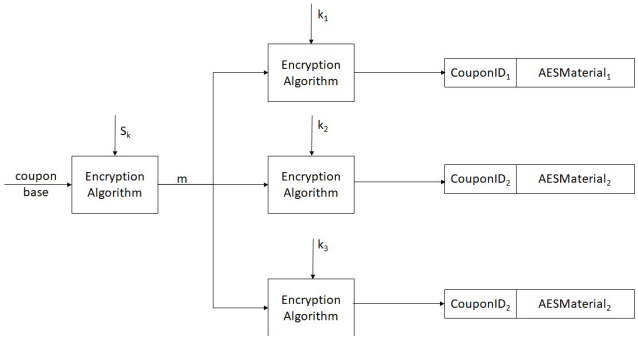

Fig. 14. Multi-Message Encryption

To facilitate the functionality, the CouponCloud calculates a mapping between coupon identifiers for the different classifications of shoppers. For three classification (e.g., bronze, silver, and gold rewards shoppers), there are three AES material sets (m_1, m_2, m_3), based on the application of rewards-level keys (k_1, k_2, k_3). To facilitate simplifying the calculation of this mapping, the coupon identity field and the AES material are treated as a continuous 20,B block of data. To utilize the system, the CouponCloud maps three level-based coupon identifiers to a single base coupon identifier. The single base identifier along with the AES base material as encrypted by the CouponCloud key is transmitted in the coupon message, as normal. However, gold-level shoppers use the gold-level key k_1, silver-level shoppers use the silver-level key k_2, and bronze-level shopper use the bronze-level key k_3 to extract the AES material from the payload of the coupon message (see Section IV-E2). Upon application of the appropriate shopper-level key, from the AES material, a new level-based coupon ID and new AES material can be extracted. The coupon ID can then be used to index into the CouponSafe as normal and the coupon can be decoded using the AES material. In this way, a different coupon is retrieved for each of the three shopper levels based on information sent in a single coupon message.

This optimization allows a dramatic decrease in the number of coupon messages that must be sent to support reward-level based coupons. Although the Aggio prototype was implemented with three levels, extending to more levels is straightforward.

V. Conclusions and Future Directions

As technology is added to our shopping experiences, the resulting solutions must protect both the shopper and the store, while still providing enhanced shopping experiences and increased sales. For managing incentives, the challenges come from the need to control the exposure of a shopper's information as well as who is given access to which coupons. Aggio tackles these challenges through the use of an encrypted CouponSafe that holds a shopper's coupons, but can only be unlocked when the store releases the keys for specific coupons. The careful design of the CouponSafe allows Aggio to use a single message to unlock different coupons in different shopper's CouponSafes. The deployment of CouponHubs in the aisles or on the shelves of a store enables localized coupon distribution to shoppers near given products. Finally, the intelligent use of encrypted messaging supports privacy preserving communication between the CouponShopper application on the shopper's mobile device and the CouponHubs, enabling the Coupon Hubs to use on demand distribution of coupons to nearby shoppers.

To continue this research, given the limited bandwidth of low-power protocols like BLE, we will further investigate Aggio's distribution scheduler in conjunction managing shopper incentives. We are also investigating the impact of shopper mobility patterns and their impact on the needed level of density of CouponHubs in the store. For deployment, we plan to deploy Aggio in various types of shopping environments, including super markets and clothing stores to investigate the impact of different types of store layouts and floor plans on the placement of and need for CouponHubs. We also want to explore time, latency, and methods for transferring the CouponSafes to the shoppers. While generally, since this delivery happens prior to shopping and the CouponSafe is relatively small, our current method does not negatively impact the shopping experience, it is worth exploring ways to optimize the creation and transmission of the CouponSafe.

Finally, now that the system is in a deployable state, we would like to institute a large user study. Such a study could spread light on such issues as: how effective is the system in a dense environment; how do shoppers locate the actual item related to the coupon (how long does it take, how far past the item are they, etc); does any user confusion result from coupon delivery; and would integration into a shopping list application benefit the system.

References

[1] Amazon.com. (2017) Amazon go stores. https://www.amazon.com/b?node=16008589011.

[2] Google Inc. (2014) Physical web. https://google.github.io/physical-web/.

[3] A. F. Harris III, V. Khanna, G. Tuncay, R. Kravets, and R. Want, "Bluetooth low energy in dense iot environments," *IEEE Communications Magazine*, pp. 30–36, Dec. 2016.

[4] G. Tuncay, V. Khanna, R. Kravets, and A. F. Harris III, "Smart vending: Iot-enabled inventory control (demo)," in *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. ACM, 2016.

[5] R. Kravets, G. S. Tuncay, and H. Sundaram, "For your eyes only," in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, ser. MCS '15. New York, NY, USA: ACM, 2015, pp. 28–35. [Online]. Available: http://doi.acm.org/10.1145/2802130.2802137

[6] Technavio. (2017) Global mobile coupons market 2016–2020. https://www.technavio.com/report/global-miscellaneous-global-mobile-coupons-market-2016-2020?utm_source=T3&utm_campaign=Media&utm_medium=BW.

[7] RetailMeNot. (2017) We are a coupon nation. http://retailmenot.mediaroom.com/2014-09-08-We-Are-a-Coupon-Nation.

[8] CreditCards.com. (2017) Card-linked offers: Shopping deals you're not aware of. https://www.creditcards.com/credit-card-news/card-linked-offers.php.

[9] Accenture. (2017) Use of smartphones by bargain–hunting consumers is changing the customer–retailer relationship, accenture survey finds. https://newsroom.accenture.com/article_display.cfm?article_id=5109.

[10] DMR. (2017) 41 interesting target statistics and facts (november 2017). https://expandedramblings.com/index.php/target-statistics/.

[11] CreditDonkey. (2017) Grocery shopping statistics: 23 fun size facts to know. https://www.creditdonkey.com/grocery-shopping-statistics.html.

[12] R. Faragher and R. Harle, "An analysis of the accuracy of bluetooth low energy for indoor positioning applications," in *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (IONGNSS+)*, 2014, pp. 201–210.

[13] A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. D. Hämäläinen, "Experiments on local positioning with bluetooth," in *Information Technology: Coding and Computing [Computers and Communications], 2003. Proceedings. ITCC 2003. International Conference on*. IEEE, 2003, pp. 297–303.

[14] A. Ghose, C. Bhaumik, and T. Chakravarty, "Blueeye: A system for proximity detection using bluetooth on mobile phones," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 1135–1142.

[15] S. Liu, Y. Jiang, and A. Striegel, "Face-to-face proximity estimation using bluetooth on smartphones," *Mobile Computing, IEEE Transactions on*, vol. 13, no. 4, pp. 811–823, 2014.

[16] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 293–304.

[17] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 173–184.

[18] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2000, pp. 775–784.

[19] I. Constandache, R. R. Choudhury, and I. Rhee, "Towards mobile phone localization without war-driving," in *Infocom, 2010 proceedings ieee*. IEEE, 2010, pp. 1–9.

[20] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 197–210.

[21] T. M. Ng, "From where i am to here i am: Accuracy study on location-based services with ibeacon technology," *HKIE Transactions*, vol. 22, no. 1, pp. 23–31, 2015.

[22] P. Martin, B.-J. Ho, N. Grupen, S. Muñoz, and M. Srivastava, "An ibeacon primer for indoor localization: Demo abstract," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ser. BuildSys '14. New York,

[28] R. Snader, R. Kravets, and A. F. Harris III, "Cryptocop: Lightweight, energy-efficient encryption and privacy for wearable devices," in *WearSys*. ACM, 2016.

NY, USA: ACM, 2014, pp. 190–191. [Online]. Available: http://doi.acm.org/10.1145/2674061.2675028

[23] A. Kwiecień, M. Maćkowski, M. Kojder, and M. Manczyk, "Reliability of bluetooth smart technology for indoor localization system," in *Computer Networks*. Springer, 2015, pp. 444–454.

[24] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: a high accuracy acoustic ranging system using cots mobile devices," in *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 2007, pp. 1–14.

[25] S. Lanzisera and K. S. Pister, "Burst mode two-way ranging with cramer-rao bound noise performance," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE, 2008, pp. 1–5.

[26] T. Sathyan, D. Humphrey, and M. Hedley, "Wasp: A system and algorithms for accurate radio localization using low-cost hardware," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 2, pp. 211–222, 2011.

[27] M. Vučinić, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, and R. Guizzetti, "OSCAR: Object security architecture for the internet of things," *Ad Hoc Netw.*, vol. 32, no. C, pp. 3–16, Sep 2015.

[29] A. F. Harris III, R. Kravets, and R. Snader, "Multicop: Extending iot privacy protections to a multi-device environment," in *SafeThings*. ACM, 2017.

[30] A. Das, P. Pathak, C.-N. Chuah, and P. Mohapatra, "Uncovering privacy leakage in ble network traffic of wearable fitness trackers," in *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. ACM, 2016.

[31] M. Ryan, "Bluetooth: With low energy comes low security," in *7th USENIX Workshop on Offensive Technologies*, 2013.

[32] P. Marquardt, D. Dagon, and P. Traynor, "Impeding individual user profiling in shopper loyalty programs."

[33] K. Partridge, M. A. Pathak, E. Uzun, and C. Wang, "PiCoDa: Privacy-preserving smart coupon delivery architecture," 2012.

[34] Apple. (2014) Getting started with ibeacon. http://apple.co/1MPb7CU.

[35] I. Google. (2017) Google cloud platform. https://cloud.google.com.

[36] ——. (2017) The go programming language. https://golang.com.

[37] G. US. (2017) Coupons: Paper-based and paperless coupon standards. https://www.gs1us.org/upcs-barcodes-prefixes/additional-ways-to-identify-products/coupons.

[38] NIST, "Federal information processing standards publication 197. Advanced Encryption Standard (AES)," 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[39] I. Nordic Semiconductor. (2017) nrf52832. Https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52832.

[40] I. Bluetooth SIG., "Bluetooth low energy 4.1 standard."

[41] Adafruit. (2017) Adafruit cc3000 wifi. Https://www.adafruit.com/product/1469.

[42] *Atmel ATAES*.

[43] (2017) Benchmarking wolfSSL and wolfCrypt. https://www.wolfssl.com/wolfSSL/benchmarks-wolfssl.html.