Efficient Anonymous Temporal-Spatial Joint Estimation at Category Level Over Multiple Tag Sets With Unreliable Channels

Youlin Zhang, *Member, IEEE*, Shigang Chen, *Fellow, IEEE*, You Zhou, *Member, IEEE*, Olufemi O. Odegbile, and Yuguang Fang, *Fellow, IEEE, Member, ACM*

***Abstract*— Radio-frequency identification (RFID) technologies have been widely used in inventory control, object tracking and supply chain management. One of the fundamental system functions is called cardinality estimation, which is to estimate the number of tags in a covered area. In this paper, we extend the research of this function in two directions. First, we perform joint cardinality estimation among tags that appear at different geographical locations and at different times. Moreover, we target at category-level information, which is more significant in practical scenarios where we need to monitor the tagged objects of many different categories. Second, we enforce anonymity in the process of information gathering in order to preserve the privacy of the tagged objects. These capabilities will enable new applications such as tracking how products of different categories are transferred in a large, distributed supply chain. We propose and implement a novel protocol to meet the requirements of anonymous category-level joint estimation over multiple tag sets. We formally analyze the performance of our estimator and determine the optimal system parameters. Moreover, we extend our protocol to unreliable channels and consider two channel error models. Extensive simulations show that the proposed protocol can efficiently and accurately estimate joint information over multiple tag sets at category level, while preserving tags' anonymity.**

***Index Terms*— Radio-frequency identification (RFID) tags, wireless application protocol, ultra high frequency (UHF) communication.**

I. Introduction

RECENT years have witnessed rapid development of RFID technologies, which has established new system functions that support numerous novel applications in logistics, inventory control, product tracking, and supply chain management [1]–[23]. In practice, each object in an area of surveillance is attached with an RFID tag, and an RFID reader is deployed with one or multiple antennas placed at chosen locations to monitor the set of tagged objects.

Cardinality Estimation: One of the fundamental functions in RFID systems is *cardinality estimation*, which is to estimate the number of tags (objects) in a surveillance area. This function has wide applications in warehouse management such as detecting management errors, theft, and vendor fraud [2], [8], [24]. It is also useful for other applications (e.g., transferring commercial goods at a port) that only require a reader to estimate the number of tagged objects, without the need of accessing the tag IDs for the purpose of keeping the anonymity of customer products. Numerous solutions [1], [3], [25]–[31] have been proposed. Comparing with the traditional approach of identifying all tag IDs and then counting the number, they consume much less time and save much energy. More importantly, time efficiency is significant for minimizing disruptions to normal operations in a busy warehouse environment [32]. One limitation of the aforementioned work is that they only consider cardinality estimation of *a single tag set* [1], [3], [26]–[30].

Multiple Tag Sets: We motivate the problem considered in this paper through an application scenario. Consider a large, distributed supply-chain network, where products are tagged and shipped from location to location over time. We want to have a tool to analyze how products flow through the network. Take a few examples: For all products shipped out from any given supplier on a certain day, how many of them are moved to each location (i.e., storage and distribution facility) on the subsequent days? On any given day, how many products are shipped from one location to another? Or more generally, how many products are shipped through a given sequence of locations? Such information can help improve the allocation of delivery resources and predict the inventories for better efficiency of the supply network.

As products are shipped in and out, the set of tags at each location changes. We consider a tag set as a spatial-temporal function of location and time, representing the set of products at a given location and a given time. The questions in the previous application scenario are all related to a fundamental *joint-estimation function* of finding the number of common tags (i.e., common products) between two or more tag sets — - for instances, the tag set for the products shipped out from a certain supply and the tag set at a given distribution facility on a later day, the tag set of an upstream distribution facility in the morning and the tag set of an downstream distribution facility in the evening, or the tag sets along a distribution chain.

One way to support the joint-estimation function is to install an RFID reader (possibly with multiple antennas for coverage) at every location to take a snapshot of the local tag set periodically after a preset time interval (e.g., an hour or a day). Here a snapshot is defined as a data structure that anonymously encodes the information of a tag set, without carrying any tag IDs; past research on cardinality estimation [1], [3], [25]–[31] has demonstrated that there are better ways of recording tag

Manuscript received December 6, 2017; revised April 25, 2019 and March 3, 2020; accepted June 22, 2020; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor Y. Guan. Date of publication July 31, 2020; date of current version October 15, 2020. This work was supported in part by the National Science Foundation under Grant CNS-1409797 and Grant CNS-1718708. *(Corresponding author: Youlin Zhang.)*

Youlin Zhang, Shigang Chen, You Zhou, and Olufemi O. Odegbile are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611-6120 USA (e-mail: youlin@cise.ufl.edu).

Yuguang Fang is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6120 USA.

Digital Object Identifier 10.1109/TNET.2020.3011347

sets that are both anonymous and much more time-efficient than collecting IDs.

Given a snapshot captured at location X in the early morning of date 1 and another snapshot captured at location Y in the evening of date 2, if we can estimate the number of common tags in the two tag sets, we will know how many products are shipped from X to Y from date 1 to date 2. By monitoring such pairs of snapshots on other dates, we will gain a good picture about how products are moved between these two locations over time. We can generalize this to three or more snapshots: Given an arbitrary number of snapshots from different locations at chosen times, the joint-estimation function estimates the number of common tags that appear in all tag sets that the snapshots represent. This function allows us to learn dynamics about the volume of products moving along a chain of locations during the times under consideration. When we apply the function to different location chains, we will gather a detailed picture about how products are moved in the whole network.

There are a few recent works studying joint estimation of *two tag sets*, i.e., estimating the cardinality of the intersection of the two sets [31], [33], [34]. Their solutions cannot be easily extended to joint estimation over *an arbitrary number of tag sets*. This more difficult problem is solved by [35], [36]. One practically important limitation is that all the above works [31], [33]–[36] can only tell us the total volume of all products moving from location to location, and cannot be zoomed in to tell the details of how each category of products flow through the network.

Multiple Categories: In the previous example of supply-chain network, each tagged object set in the location chains may consist of numerous types of products. Practically, knowing how each type of products flows through the network is much more useful than knowing the total number of products shipped from one location to another. To support product types, we put tags into categories, one category for each product type, with all tags in the same category sharing a common category ID. Given an arbitrary number of snapshots from different locations at chosen times, the problem of joint estimation at category level is to estimate the number of *common tags in each category*, which appear in all the snapshots; recall that each snapshot represents a tag set and does not carry any tag (category) ID information. That is, we want to anonymously estimate the cardinality of the intersection over multiple tag sets for each category.

As mentioned earlier, [35], [36] can estimate the cardinality of the intersection of multiple tag sets. One may suggest that we apply them repetitively, one category at a time, to obtain category-level information. This can be done by the reader announcing one category ID each time so that only tags matching that category ID will respond. Such an approach however breaks the anonymity of category IDs (which may be more important than individual object IDs because they reveal the product types). To make things even worse, the approach is inefficient as we will demonstrate in this paper.

There is very limited prior work that supports tag categories. Related is the work that classifies the categories in a single tag set [32], [37]. The only work that performs category-level joint estimation [38] can deal with only two tag sets, and its analytical framework cannot be easily extended to more sets. This paper proposes a new protocol for anonymous joint estimation over multiple tag sets at category level. To keep the anonymity of tags, we thoroughly mix the information from tags of all categories in one snapshot, without revealing any ID information during information gathering. In this way, even if unauthorized readers are able to eavesdrop on the communication, they cannot easily obtain any tag/category ID(s). To perform joint estimation, we first combine the snapshots, during which two kinds of noises are introduced: inter-set noise and inter-category noise. Then we use statistical methods to estimate and remove the noise (due to mixing) in the combination to obtain the cardinality of the intersection of multiple tag sets for each category.

The main contributions of this paper are summarized as follows:

First, we extend the traditional RFID estimation problem to more practical scenarios where we perform joint category-level estimation over multiple tag sets at different locations and/or different times. Not only is the problem more challenging, but the proposed solution allows us to learn the spatial-temporal dynamics among these tag sets and their associated objects.

Second, we enforce anonymity in the proposed multi-set category-level tag estimation protocol.

Third, we formally analyze the performance of our protocol. Through statistical analysis, we show that our estimator for category-level joint information is asymptotically unbiased and can be made to meet any preset estimation accuracy requirement. Our simulations show that the proposed protocol can efficiently and accurately obtain category-level estimation, while preserving tags' anonymity.

Finally, we extend our protocol to make it work under unreliable channels and investigate the impact of different channel errors.

The rest of our paper is organized as follows. Section II presents our system model and formally defines the problem of category-level joint estimation. Section III presents and analyzes our new protocol for joint estimation. Section IV extends our solution under unreliable channels. Section V discusses the implementation issues. Section VI evaluates our protocol through simulations. Section VII discusses the related work. Section VIII draws the conclusion.

II. System Model and Problem Definition

A. System Model

Consider a large distributed RFID system, where all the objects are classified into m different categories with a set M of *category* IDs, $\{cid_1, cid_2, \ldots, cid_m\}$. Each object is attached with a tag and can be uniquely identified by a *tag* ID id, which contains a category ID cid and an object ID tid, with the former specifying which category the object belongs to and the latter being unique in the same category. Typically, the length of a tag ID is 96 bits. If we use 16 bits for category IDs, they can support 65,536 different categories, which is sufficient for a large RFID-assisting supply-chain network with tens of thousands of suppliers.

The reader initializes communication with tags by broadcasting a request, which contains all necessary parameters including a frame size and random seeds. Each tag after receiving the request will choose a slot based on the random seeds it receives and transmit a tag response in that slot. The reader will produce a snapshot of the tag set based on the responses sent back from the tags. We will discuss the structure

of the snapshot and the implementation of the communication protocol later.

Suppose unauthorized adversaries may plant readers at chosen locations to eavesdrop on the communication between tags and readers, from which they try to infer private information such as tag IDs and category IDs about the products. We assume that the adversaries do not have prior knowledge of the tag IDs or category IDs in the system. Our goal is to prevent them from acquiring any tag ID(s).

B. Problem Definition

Given k snapshots captured at different locations or at the same location but different times. We denote them as $B_1, B_2, \ldots, B_k$ and the tag sets that they represent as $T_1, T_2, \ldots, T_k$, respectively. Let C_i^{cid} be the subset of tags in T_i that belong to a category $cid \in M$. Clearly, $T_i = C_i^{cid_1} \cup C_i^{cid_2} \ldots \cup C_i^{cid_m}$. We will study the joint property of the k tag sets for each category in M.

Let $C_*^{cid} = C_1^{cid} \cap C_2^{cid} \ldots \cap C_k^{cid}$ and $n_*^{cid} = |C_*^{cid}|$, where the subscript $*$ means the *common tags* among the k subsets. Because all operations are applied to each category independently and separately, in the sequel we will leave out the superscript cid in operation description to simplify the notations. We abbreviate C_i^{cid}, C_*^{cid} and n_*^{cid} as C_i, C_* and n_* respectively.

The problem of anonymous category-level joint estimation over multiple tag sets is to estimate n_* as $\hat{n}_*$ for each category with (i) an accuracy requirement,

$$Prob\{|\hat{n}_* - n_*| \leq e\} \geq \alpha. \tag{1}$$

where e is an absolute error bound and α is a constant, which is referred as the *confidence level*, and (ii) an anonymous requirement that no tag ID or category ID is transmitted during the estimation process.

As an example for the accuracy requirement, if $\alpha = 95\%$ and $e = 50$, we require that the probability for the estimation error $|\hat{n}_* - n_*|$ to be bounded by 50 is at least 95%.

The prior work [1], [3], [25], [26], [28] on cardinality estimation of a *single* tag set adopts a relative error model:

$$Prob\{|\hat{n}_* - n_*| \leq \varepsilon n_*\} \geq \alpha. \tag{2}$$

where ε is the relative error requirement. However, for joint estimation over multiple sets, the execution time is inversely related to the Jaccard similarity, $J = \frac{n_*}{n}$, where n_* is the number of common tags in the tag sets and n is the total number of tags in all sets. For example, the time complexity of [35] is $\Theta(\frac{1}{\varepsilon^2 J} \ln \frac{1}{1-\alpha})$, under the relative error model. While n is typically big for a large RFID system, the value of n_* can be large or small, even down to zero, causing the term $\frac{1}{J}$ to approach infinity. That is the reason why the more recent work of [36] advocates the absolute error model (1), which we adopt in this paper. For the prior work [1], [3], [25], [26], [28] on a single set, their Jaccard similarity is one since $n = n^*$. Therefore, the relative error model is fine.

Because the communication channel between the reader and tags is open to eavesdroppers, we add an anonymous feature to our protocol design such that tag/category IDs are not transmitted explicitly by tags or the reader.

Table I lists some frequently used notations, which we will adopt later in our description.

TABLE I

FREQUENTLY USED NOTATIONS. A NOTATION FOR AN ESTIMATED VALUE WILL CARRY A HAT. FOR EXAMPLE, $\hat{n}_i$ IS THE ESTIMATED VALUE FOR n_i. THE FOLLOWING SYMBOLS ACTUALLY HAVE cid AS SUPERSCRIPT: n_i, $n_{c_1\,c_2 \ldots c_i}$, n_*, V, U, $U_{c_1\,c_2 \ldots c_i}$ AND $V_{c_1\,c_2 \ldots c_i}$. WE OMIT THE SUPERSCRIPT FOR CLARITY WHEN THE CONTEXT DOES NOT RAISE CONFUSION

Notation	Description
k	number of tag sets
c_i	id of the ith tag set, $1 \leq i \leq k$
t_i	number of tags in the ith tag set under consideration
$t_{c_1 c_2 \ldots c_i}$	total number of tags among tag set c_1, c_2, ..., c_i
m	number of categories
cid	id of an arbitrary category in a tag set
n_i	number of tags from category cid in the ith tag set
$n_{c_1 c_2 \ldots c_i}$	total number of tags from category cid among tag set c_1, c_2, ..., c_i
n_*	number of common tags from category cid among k tag sets
f	size of the snapshot
l	size of the virtual snapshot
V	fraction of zero bits in a virtual bitmap
U	fraction of zero bits in a bitmap
$U_{c_1 c_2 \ldots c_i}$	fraction of zero bits in a combined virtual bitmap
$V_{c_1 c_2 \ldots c_i}$	fraction of zero bits in a combined bitmap

C. Performance Metrics

We use three metrics for performance evaluation.

1) Estimation Accuracy: The accuracy requirement is specified in (1).

2) Execution Time: Since RFID systems operate in low-rate communication channels, time efficiency is an important performance metric, especially when the number of tags is very large. Therefore, it is imperative that the design of a protocol can reduce execution time as much as possible. In this paper, we adopt the number of time slots needed by each protocol as the metrics for execution time.

3) Anonymity: We use two probability values, p_{id} and p_{cid}, to quantify the preserved anonymity of tag IDs and category IDs, respectively. More specifically, p_{id} (p_{cid}) is the probability that any tag (category) ID is not revealed to an eavesdropper that listens to all wireless communications. In practice, we want to make p_{id} and p_{cid} as close to 1 as possible.

III. EFFICIENT ANONYMOUS TEMPORAL-SPATIAL JOINT ESTIMATION AT CATEGORY LEVEL OVER MULTIPLE TAG SETS

In this section, we present our new protocol for efficient anonymous temporal-spatial Joint Estimation at Category level over Multiple tag sets (JECM). JECM consists of two phases: an online encoding phase and an offline analysis phase. During the online encoding phase, each tag set that resides at a certain location and a certain time in the system is encoded into a snapshot by the reader. During the offline analysis phase, all snapshots are loaded to a central server where joint estimation is performed. We adopt an asymmetric design that pushes most complexity to the reader as well as the central server, while keeping the tag operation simple. The only thing that a tag needs to do is to make a single transmission in response to a reader's request during online encoding.

A. Structure of Snapshot

Consider an arbitrary tag set T_i at a certain location and a certain time in a large distributed RFID system. To take a

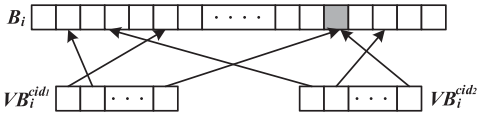


Fig. 1. An illustration of drawing bits randomly from a bitmap B_i to construct virtual bitmaps $VB_i^{cid_1}$ and $VB_i^{cid_2}$. The bit in grey is shared by both virtual bitmaps.

snapshot, the reader broadcasts a request, which is followed by a slotted ALOHA frame. Upon receipt of the request, each tag pseudo-randomly selects a slot in the frame and sends back a signal response in the slot. The reader monitors the status of each slot, which is referred to either as an *empty slot* where no tag responds or as a *busy slot* where one or more tags respond. The reader converts the time frame into a bitmap B_i, zero for each empty slot and one for each busy slot. We use B_i as the snapshot of T_i.

Encoding category-level information is tricky. Establishing one bitmap for each category is problematic. As discussed in Section I, this requires the reader to broadcast one request per category, carrying a category ID to ask only tags in the category to respond, which breaks anonymity. Moreover, this approach takes long execution time as is demonstrated in Section VI. Our idea is to mix information from all categories in the same bitmap to improve anonymity and time efficiency as no category ID will be transmitted and it takes just one request-response round to build the snapshot for all categories. To do so, we must introduce additional structure to the snapshot B_i. For each category cid, we pseudo-randomly select a certain number l of bits from B_i to encode tags of that category. Logically, these bits form a *virtual bitmap* VB_i^{cid}. Fig. 1 illustrates the idea of drawing bits from B_i to form two virtual bitmaps, $VB_i^{cid_1}$ and $VB_i^{cid_2}$, for two categories cid_1 and cid_2, respectively. Different categories may share bits in B_i due to random bit selection, which brings two benefits: First, information from different categories is mixed, which is good for anonymity. Second, it improves time efficiency. The value of l has to be set reasonably large so that there are sufficient bits to encode large categories. If separate bits were designated for different categories, many bits for small categories may be left unused. Thanks to random sharing, in our design, the unused bits for small categories can be picked up by other categories, which reduces the total number of bits (time slots) needed.

The challenge is how to take a snapshot with embedded category structure and how to perform accurate joint estimation when category level information is mixed.

B. Online Encoding

Denote the jth bit in the bitmap as $B_i[j]$, $0 \le j \le f-1$. Consider an arbitrary category cid, whose jth bit is denoted as $VB_i^{cid}[j]$, $0 \le j \le l-1$. Since our discussion is involved only a single category ID, we will leave out the superscript cid. The selection of $VB_i[j]$ from B_i is formally defined as

$$VB_i[j] \equiv B_i[H_j(cid)]. \tag{3}$$

where $H_j()$ is a hash function. Instead of requiring l different hash functions, we implement them based on a common master hash function $H(\ldots)$ and l different random seeds, $r_0, r_1, \ldots, r_{l-1}$,

$$H_j(cid) = H(cid \oplus r_j), \quad 0 \le j \le l-1. \tag{4}$$

For online encoding, the reader broadcasts a request, which includes the frame size f and l random seeds. The request is followed by an ALOHA frame of f slots. Consider an arbitrary tag. Without losing generality, suppose the tag belongs to category cid. It should be encoded by one of the bits in the category's virtual bitmap VB_i. The tag uses another hash function $h(tid) \in [0, l-1]$ to choose a bit pseudo-randomly from VB_i, where $h()$ may also be implemented using the master hash function with a pre-defined seed. By (3), the bit $VB_i[h(tid)]$ is actually the $H_{h(tid)}(cid)$th bit in B_i, which corresponds to the $H_{h(tid)}(cid)$th slot in the time frame. Hence, the tag will choose that time slot to transmit. Once the reader finds the $H_{h(tid)}(cid)$th slot is busy, it sets

$$B_i[H_{h(tid)}(cid)] = 1. \tag{5}$$

C. Offline Category-Level Joint Estimation Over Multiple Tag Sets

With online encoding, snapshots of different tag sets are sent to a central server via the reader. Consider joint estimation over an arbitrary set of k snapshots, $\{B_1, B_2, \ldots, B_k\}$. There are $2^k - 1$ subsets of $\{B_1, B_2, \ldots, B_k\}$, excluding the empty subset. Consider an arbitrary subset $\{B_{c_1}, B_{c_2}, \ldots, B_{c_i}\}$, where $1 \le i \le k$ and $1 \le c_1 < c_2 < \ldots < c_i \le k$. The server combines the bitmaps in the subset by bitwise OR. Namely, the combined bitmap $B_u^{c_1\ c_2 \cdots c_i}$ is defined as

$$B_u^{c_1 c_2 \cdots c_i} = B_{c_1} \vee B_{c_2} \vee \ldots \vee B_{c_i}. \tag{6}$$

where $\vee$ is the bitwise-OR operation. As a result, the information from tags in different bitmaps is combined. For example, B_u^{12} is the combined bitmap of B_1 and B_2. The combined bitmaps will be used later in our estimation.

The server retrieves per-category information from the k tag sets by reconstructing the bitmap VB_i (short for VB_i^{cid}) as follows:

$$VB_i[j] \equiv B_i[H_j(cid)], \quad 1 \le i \le k,\ 0 \le j \le l-1. \tag{7}$$

Again, there are $2^k - 1$ subsets of $\{VB_1, VB_2, \ldots, VB_k\}$, excluding the empty subset. Consider an arbitrary subset $\{VB_{c_1}, VB_{c_2}, \ldots, VB_{c_i}\}$, where $1 \le i \le k$ and $1 \le c_1 < c_2 < \ldots < c_i \le k$. The server constructs the combined virtual bitmap $VB_u^{c_1\ c_2 \cdots c_i}$ as

$$VB_u^{c_1 c_2 \cdots c_i} = VB_{c_1} \vee VB_{c_2} \vee \ldots \vee VB_{c_i}. \tag{8}$$

Fig. 2 shows an example of how to construct B_u^{123} and VB_u^{123} from subsets $\{B_1, B_2, B_3\}$ and $\{VB_1, VB_2, VB_3\}$, respectively.

For each snapshot, virtual bitmaps for all categories share the bits in the same underlying bitmap B_i, $1 \le i \le k$. A bit "1" in VB_i may not be set by tags belonging to category cid, but instead be set by tags of other categories, resulting false positives in virtual bitmaps. When bitmaps and virtual bitmaps are combined, a bit in $B_u^{c_1\ c_2 \cdots c_i}$ or $VB_u^{c_1\ c_2 \cdots c_i}$ may be set to one by tags from different sets, which also introduces false positives. These false positives are considered as inter-set and inter-category noises and we must remove them in deriving our estimation formula.

We use probabilistic methods to analyze the expected fraction of zero bits in all the bitmaps and virtual bitmaps we obtain and derive the estimator $\hat{n}_*$ for n_*.

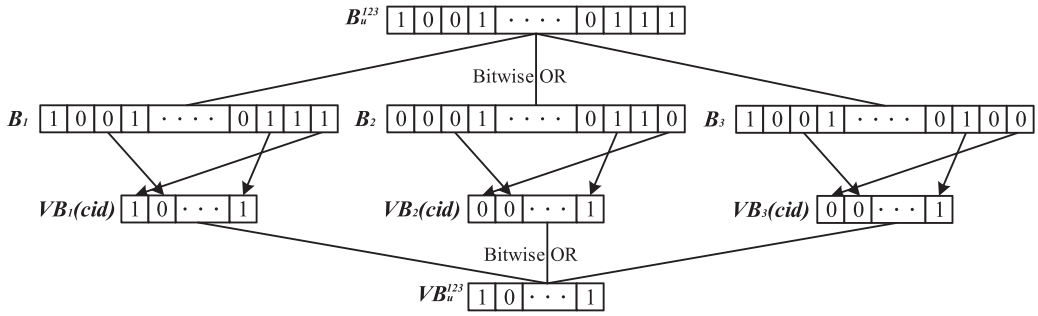


Fig. 2. An illustration of how to construct B_u^{123} and VB_u^{123} from subsets $\{B_1, B_2, B_3\}$ and $\{VB_1, VB_2, VB_3\}$, respectively.

Before we continue our derivation of $\hat{n}_*$, we want to first present the basis for our estimator. In set theory, the cardinality of the intersection of k tag sets can be derived based on the well-known *principle of inclusion and exclusion*:

$$\begin{aligned}|C_1 \cap C_2 \ldots \cap C_k| &= \sum_{1\le c_1\le k} |C_{c_1}| - \sum_{1\le c_1<c_2\le k} |C_{c_1} \cup C_{c_2}| + \ldots \\ &+ (-1)^{i+1} \sum_{1\le c_1 \ldots < c_i \le k} |C_{c_1} \cup C_{c_2} \ldots \cup C_{c_i}| \\ &+ (-1)^{k+1} |C_1 \cup C_2 \cup \ldots \cup C_k|. \end{aligned} \tag{9}$$

where $C_{c_1} \cup C_{c_2} \ldots \cup C_{c_i}$ is the union of i tag sets and $|C_1 \cap C_2 \ldots \cap C_k|$ is the category-level joint information of k sets we want to estimate. From equation (9), we can observe that in order to obtain the joint information of k tag sets, we need to compute the cardinality of all $2^k - 1$ combined sets first. JECM uses the $2^k - 1$ bitmaps obtained in Subsection III-C to estimate the cardinality of each combined set and finally obtains category-level joint information of k tag sets by using the principle of inclusion and exclusion. Next we will show how we derive the cardinality of each combined set.

We start with estimating the cardinality of C_i in an arbitrary tag set T_i using bitmaps B_i and VB_i. Consider a tag set T_i. Since we now focus on one tag set, in the sequel, we will leave out set index i for B_i, VB_i, T_i and C_i. Remember we also leave out the superscript cid in C_i as mentioned in Subsection II-B. We denote t as the number of tags in T (all tags of all categories) and n as the number of tags in C (tags belonging to category cid).

For an arbitrary bit b in an f-bit bitmap B, a tag t has a probability $\frac{1}{f}$ to set it as one and we denote U as the fraction of zero bits in B. Let $\mathcal{A}_j$ be the event that the jth $0 \le j \le f-1$ bit in B remains zero after online encoding, and 1_{A_j} be the corresponding indicator random variable, that is,

$$1_{\mathcal{A}_j} = \begin{cases} 1, & \text{if } B[j] = 0, \\ 0, & \text{if } B[j] = 1. \end{cases}$$

So we have $U = \frac{\sum_{j=0}^{f-1} 1_{\mathcal{A}_j}}{f}$ and $P(\mathcal{A}_j) = (1 - \frac{1}{f})^t$. Therefore,

$$\begin{aligned} E(U) &= \frac{1}{f} \sum_{j=0}^{f-1} E(1_{\mathcal{A}_j}) \\ &= \frac{1}{f} \sum_{j=0}^{f-1} [1 \times P(\mathcal{A}_j) + 0 \times (1 - P(\mathcal{A}_j)] \\ &= (1 - \frac{1}{f})^t. \end{aligned} \tag{10}$$

Now we will move on to investigate the properties of a virtual bitmap VB. We denote V as the fraction of zero bits in VB. Let $\mathcal{B}_j$ be the event that the jth $0 \le j \le l-1$ bit in VB remains 0 after online encoding, and $1_{\mathcal{B}_j}$ be the corresponding indicator random variable. Similarly,

$$1_{\mathcal{B}_j} = \begin{cases} 1, & \text{if } VB[j] = 0, \\ 0, & \text{if } VB[j] = 1. \end{cases}$$

In this condition, in order to make a bit in VB remain zero, neither the tags in category cid nor the tags belong to other categories shall set $VB[j]$. The probability for a tag in category cid not to set $VB[j]$ is $(1 - \frac{1}{l})^n$ and the probability for a tag belonging to other categories not to set $VB[j]$ is $(1 - \frac{1}{f})^{t-n}$. Thus, we have $P(\mathcal{B}_j) = (1 - \frac{1}{l})^n (1 - \frac{1}{f})^{t-n}$ and the expected value of V can be derived as:

$$\begin{aligned} E(V) &= \frac{1}{l} \sum_{j=0}^{l-1} E(1_{\mathcal{B}_j}) \\ &= \frac{1}{l} \sum_{j=0}^{l-1} [1 \times P(\mathcal{B}_j) + 0 \times (1 - P(\mathcal{B}_j)] \\ &= (1 - \frac{1}{l})^n (1 - \frac{1}{f})^{t-n}. \end{aligned} \tag{11}$$

Combining (10) with (11), we have

$$E(V) = (1 - \frac{1}{l})^n (1 - \frac{1}{f})^{-n} E(U). \tag{12}$$

Substituting $E(U)$ and $E(V)$ with U and V respectively, and taking a logarithm on both sides, we derive an estimator for n as:

$$\hat{n} = \frac{\ln V - \ln U}{\ln(1 - \frac{1}{l}) - \ln(1 - \frac{1}{f})}. \tag{13}$$

Recall that we leave out the tag set index i, as well as category id cid in all these formulas. As a result, we obtain the category-level cardinality information $\hat{n}_i^{cid}$ for each tag set.

Now we will investigate the properties of combined bitmaps $B_u^{c_1\ c_2 \ldots c_i}$ and $VB_u^{c_1\ c_2 \ldots c_i}$, $1 \le i \le k$.

Let $\mathcal{C}_j$ be the event that jth bit in $B_u^{c_1\ c_2 \ldots c_i}$ remains zero after online encoding and $U_{c_1\ c_2 \ldots c_i}$ be the fraction of zeros in $B_u^{c_1\ c_2 \ldots c_i}$. We denote $1_{\mathcal{C}_j}$ as the indicator random variable of $\mathcal{C}_j$. Since $B_u^{c_1\ c_2 \ldots c_i}$ is the combination of i tag sets $c_1, c_2, \ldots,$ and c_i, z will remain zero if and only if z is not chosen by any tag in these i sets, that is,

$$P(\mathcal{C}_j) = (1 - \frac{1}{f})^{t_{c_1 c_2 \ldots c_i}}. \tag{14}$$

where $t_{c_1\ c_2\ldots c_i}$ is the number of tags in all these i tag sets and $t_{c_1\ c_2\ldots c_i} = |T_{c_1} \cup T_{c_2} \ldots \cup T_{c_i}|$. Therefore, the expected value of $U_{c_1\ c_2\ldots c_i}$ can be computed as:

$$\begin{aligned} E(U_{c_1\ c_2\ldots c_i}) &= \frac{1}{f}\sum_{j=0}^{f-1} E(1_{\mathcal{C}_j}) \\ &= \frac{1}{f}\sum_{j=0}^{f-1}[1 \times P(\mathcal{C}_j) + 0 \times (1 - P(\mathcal{C}_j)] \\ &= (1-\frac{1}{f})^{t_{c_1 c_2\ldots c_i}}. \end{aligned} \tag{15}$$

For a combined virtual bitmap $VB_u^{c_1\ c_2\ldots c_i}$, let $\mathcal{D}_j$ be the event that the jth bit in $VB_u^{c_1\ c_2\ldots c_i}$, and $1_{\mathcal{D}_j}$ be the corresponding indicator random variable. In this situation, $1_{\mathcal{D}_j}$ will be true only if the following two conditions are satisfied:

1) The jth bit is not picked by any tag in $C_{c_1} \cup C_{c_2} \ldots \cup C_{c_i}$.

2) The jth bit is not picked by any tag in $(T_{c_1} \cup T_{c_2} \ldots \cup T_{c_i}) - (C_{c_1} \cup C_{c_2} \ldots \cup C_{c_i})$.

For the first condition, the probability q_1 for it to be satisfied is given by

$$q_1 = (1-\frac{1}{l})^{n_{c_1 c_2\ldots c_i}}. \tag{16}$$

where $n_{c_1\ c_2\ldots c_i}$ is the number of tags belonging to category cid in all k sets and $n_{c_1\ c_2\ldots c_i} = |C_{c_1} \cup C_{c_2} \ldots \cup C_{c_i}|$. Similarly, the probability q_2 for the second condition to be satisfied is given by

$$q_2 = (1-\frac{1}{f})^{t_{c_1 c_2\ldots c_i} - n_{c_1 c_2\ldots c_i}}. \tag{17}$$

Combining (16) and (17), we have

$$\begin{aligned} P(\mathcal{D}_j) &= q1 \times q2 \\ &= (1-\frac{1}{l})^{n_{c_1 c_2\ldots c_i}}(1-\frac{1}{f})^{t_{c_1 c_2\ldots c_i} - n_{c_1 c_2\ldots c_i}}. \end{aligned} \tag{18}$$

Let $V_{c_1\ c_2\ldots c_i}$ be the fraction of zeros in $VB_u^{c_1\ c_2\ldots c_i}$ and the expected value can be derived as

$$\begin{aligned} E(V_{c_1\ c_2\ldots c_i}) &= \frac{1}{l}\sum_{j=0}^{l-1} E(1_{\mathcal{D}_j}) \\ &= (1-\frac{1}{l})^{n_{c_1 c_2\ldots c_i}}(1-\frac{1}{f})^{t_{c_1 c_2\ldots c_i} - n_{c_1 c_2\ldots c_i}}. \end{aligned} \tag{19}$$

Apply (15) to (19)

$$E(V_{c_1 c_2\ldots c_i}) = (1-\frac{1}{l})^{n_{c_1 c_2\ldots c_i}}(1-\frac{1}{f})^{-n_{c_1 c_2\ldots c_i}} E(U_{c_1 c_2\ldots c_i}). \tag{20}$$

Substitute $E(U_{c_1\ c_2\ldots c_i})$, $E(V_{c_1\ c_2\ldots c_i})$ with the observed value $U_{c_1\ c_2\ldots c_i}$, $V_{c_1\ c_2\ldots c_i}$ respectively, take a logarithm on both sides, and the estimator for $n_{c_1\ c_2\ldots c_i}$ can be derived as:

$$\hat{n}_{c_1 c_2\ldots c_i} = \frac{\ln V_{c_1 c_2\ldots c_i} - \ln U_{c_1 c_2\ldots c_i}}{\ln(1-\frac{1}{l}) - \ln(1-\frac{1}{f})}. \tag{21}$$

Combining (9), (13) and (21), we have our estimator $\hat{n}_*$ as

$$\hat{n}_* = \frac{\sum_{i=1}^{k}[(-1)^{i+1} \sum_{1\le c_1<\ldots<c_i\le k} (\ln V_{c_1 c_2\ldots c_i} - \ln U_{c_1 c_2\ldots c_i})]}{\ln(1-\frac{1}{l}) - \ln(1-\frac{1}{f})}. \tag{22}$$

D. Mean and Variance of $\hat{n}_*$

In this section, we analyze the statistical properties, mean and variance of $\hat{n}_*$.

In order to derive the mean and variance of $\hat{n}_*$, we need to first derive the mean and variance of $-\ln U_{c_1\ c_2\ldots c_i}$ and $-\ln V_{c_1\ c_2\ldots c_i}$. Let $\hat{u}_{c_1\ c_2\ldots c_i} = -\ln U_{c_1\ c_2\ldots c_i}$ and $\hat{v}_{c_1\ c_2\ldots c_i} = -\ln V_{c_1\ c_2\ldots c_i}$.

In [39], K. Whang *et al.* use Taylor expansion to derive the mean and variance of $\hat{u}_{c_1\ c_2\ldots c_i}$ and the results are given by:

$$E(\hat{u}_{c_1 c_2\ldots c_i}) = \frac{1}{f}(t_{c_1 c_2\ldots c_i} + \frac{e^\omega - \omega - 1}{2}), \tag{23}$$

$$Var(\hat{u}_{c_1 c_2\ldots c_i}) = \frac{1}{f}(e^\omega - \omega - 1). \tag{24}$$

where $t_{c_1\ c_2\ldots c_i}$ is the number of tags in all these i tag sets and $\omega = \frac{t_{c_1\ c_2\ldots c_i}}{f}$. Usually the frame size f is chosen such that ω is very small and $(e^\omega - \omega - 1)$ is negligible when compared to $t_{c_1\ c_2\ldots c_i}$. In this case, we will have $E(\hat{u}_{c_1\ c_2\ldots c_i}) \simeq \frac{t_{c_1\ c_2\ldots c_i}}{f}$ and the standard derivation, which is the root of $Var(\hat{u}_{c_1\ c_2\ldots c_i})$ will also be insignificant compared to the mean.

Next we derive the mean and variance of $\hat{v}_{c_1\ c_2\ldots c_i}$. In [40], M. Yoon *et al.* use Taylor expansion and statistical methods to derive the mean and variance of $\hat{v}_{c_1\ c_2\ldots c_i}$ and the results are given by:

$$E(\hat{v}_{c_1 c_2\ldots c_i}) = \alpha + \frac{e^\alpha - \omega' - 1}{2l}, \tag{25}$$

$$Var(\hat{v}_{c_1 c_2\ldots c_i}) = \frac{1}{l}(e^\alpha - \omega' - 1). \tag{26}$$

where $\alpha = \frac{t_{c_1\ c_2\ldots c_i} - n_{c_1\ c_2\ldots c_i}}{f} + \frac{n_{c_1\ c_2\ldots c_i}}{l}$, $\omega' = \frac{n_{c_1\ c_2\ldots c_i}}{l}$, and $n_{c_1\ c_2\ldots c_i}$ is the number of tags belonging to category cid in all k sets. Similarly, if l is large enough, we can obtain $E(\hat{v}_{c_1\ c_2\ldots c_i}) \simeq \alpha$.

Combining (23) and (25), we have:

$$\begin{aligned} E(\ln V_{c_1 c_2\ldots c_i} - \ln U_{c_1 c_2\ldots c_i}) &= E(\hat{u}_{c_1 c_2\ldots c_i}) - E(\hat{v}_{c_1 c_2\ldots c_i}) \\ &\simeq \frac{t_{c_1 c_2\ldots c_i}}{f} - \alpha \\ &= \frac{n_{c_1 c_2\ldots c_i}}{f} - \frac{n_{c_1 c_2\ldots c_i}}{l}. \end{aligned} \tag{27}$$

Thus, the mean of $\hat{n}_*$ can be derived as:

$$\begin{aligned} E(\hat{n}_*) &= \frac{\sum_{i=1}^{k}[(-1)^{i+1} \sum_{1\le c_1<\ldots<c_i\le k} E(\ln V_{c_1\ldots c_i} - \ln U_{c_1\ldots c_i})]}{\ln(1-\frac{1}{l}) - \ln(1-\frac{1}{f})} \\ &\simeq \frac{\sum_{i=1}^{k}[(-1)^{i+1} \sum_{1\le c_1<\ldots<c_i\le k} E(\ln V_{c_1\ldots c_i} - \ln U_{c_1\ldots c_i})]}{\frac{1}{f} - \frac{1}{l}} \\ &\simeq \sum_{i=1}^{k}[(-1)^{i+1} \sum_{1\le c_1<\ldots<c_i\le k} n_{c_1 c_2\ldots c_i}] \\ &= n_*. \end{aligned} \tag{28}$$

Similarly, the variance of $\hat{n}_*$ can be calculated as:

$$Var(\hat{n}_*) = \frac{Var(\sum_{i=1}^{k}[(-1)^{i+1} \sum_{1\le c_1<\ldots<c_i\le k} (\hat{u}_{c_1\ldots c_i} - \hat{v}_{c_1\ldots c_i})])}{r^2}. \tag{29}$$

where r is a constant and $r = \ln(1-\frac{1}{l}) - \ln(1-\frac{1}{f})$.

TABLE II
PRESERVED ANONYMITY OF DIFFERENT PROTOCOLS

Protocol	p_{cid}	p_{id}
CCF	0	$1-\frac{f}{2^{l_{id}-l_{cid}}}$
MJREP	0	$1-\frac{f}{2^{l_{id}-l_{cid}}}$
JECM	$1-\frac{f}{l\cdot 2^{l_{cid}}}$	$1-\frac{f}{2^{l_{id}}}$

In order to derive $Var(\hat{n}_*)$, we need to calculate $Var(\hat{u}_{c_1\ c_2\ldots c_i})$, $Var(\hat{v}_{c_1\ c_2\ldots c_i})$ and the covariance of $\hat{u}_{c_1\ c_2\ldots c_i}$ and $\hat{v}_{c_1\ c_2\ldots c_i}$. The covariance can also be derived using Taylor expansion, which is similar to the process in [39], [40]. As [40] shows, the covariance of $\hat{u}_1$ and $\hat{u}_{12}$ can be approximated as:

$$\begin{aligned} &Cov(\hat{u}_1,\hat{u}_{12})\\ &\quad= E(\hat{u}_1\hat{u}_{12})-E(\hat{u}_1)E(\hat{u}_{12})\\ &\quad= -E(\hat{u}_1)E(\hat{u}_{12})-\ln E(\hat{u}_1)\ln E(\hat{u}_{12})\\ &\qquad+\ln E(U_1)E(\hat{u}_{12})+E(\hat{u}_1)\ln E(U_{12}). \end{aligned} \tag{30}$$

Substituting the formula of $E(U_1)$, $E(U_{12})$, $E(\hat{u}_1)$ and $E(\hat{u}_{12})$, which we have obtained already, we can obtain the covariance. After obtaining all the covariances and variances in (29), we can calculate the $Var(\hat{n}_*)$.

E. Analysis of Anonymity

In this section, we analyze the preserved anonymity of a tag while executing our protocol. Let l_{id} and l_{cid} be the length of tag IDs and category IDs (in binary), respectively. Since the unauthorized adversary does not have any prior knowledge of the tag IDs or category IDs in our system, it can only speculate one tag ID or category ID for each slot it eavesdrops on. Therefore, the anonymity of our protocol can be characterized by the probability that the adversary identifies the correct tag ID or category ID.

1) Anonymity of Category IDs: For an l_{cid}-bit category ID, there are $2^{l_{cid}}$ possible category IDs. Each category is assigned into an l-bit virtual bitmap drawn from an f-bit bitmap. As a result, each bit in the bitmap will correspond to an average of $\frac{l\cdot 2^{l_{cid}}}{f}$ categories. Since an adversary does not have any prior information about any categories that are mapped to the same slot, the probability for the adversary to infer the correct category ID of a tag is $\frac{1}{\frac{l\cdot 2^{l_{cid}}}{f}}=\frac{f}{l\cdot 2^{l_{cid}}}$. Therefore, the anonymity of a category ID for JECM, namely p_{cid}, as $p_{cid}=1-\frac{f}{l\cdot 2^{l_{cid}}}$.

2) Anonymity of Tag IDs: For an l_{id}-bit tag ID, the bits that are available for an object ID tid are $(l_{id}-l_{cid})$-long. As a result, there are $2^{(l_{id}-l_{cid})}$ possible object IDs per category. Meanwhile, each tag belonging to the same category is randomly assigned to a slot in an l-bit virtual bitmap. Hence, the average number of tags that are mapped to one slot in the same virtual bitmap is $\frac{2^{l_{id}-l_{cid}}}{l}$. According to Section III-E.1, the adversary has a probability of $\frac{f}{l\cdot 2^{l_{cid}}}$ to infer the correct category id of a tag. Thus, the probability for the adversary to infer the correct tag ID is $\frac{f}{l\cdot 2^{l_{cid}}}\times\frac{l}{2^{l_{id}-l_{cid}}}=\frac{f}{2^{l_{id}}}$. As a result, the anonymity of a tag ID for JECM is given as $p_{id}=1-\frac{f}{2^{l_{id}}}$.

Table II shows the preserved anonymity of different protocols when performing category-level joint estimation of multiple tag sets. As we can see from this table, only our JECM protocol can preserve category anonymity p_{cid} and tag anonymity p_{id} simultaneously, while CCF and MJREP cannot preserve category anonymity. In terms of tag anonymity, JECM is the highest among three protocols when frame sizes are the same among them.

F. Parameter Setting

In order to reduce the execution time of our protocol, we optimize the parameters f and l in JECM protocol under the accuracy constraints given in (1). In Subsection III-D, we prove that $\hat{n}_*$ is asymptotically unbiased and is approximately distributed with Gaussian distribution. For a Gaussian distribution with $E(\hat{n}_*)\simeq\hat{n}_*$, equation (1) can be translated to

$$Var(\hat{n}_*)\le(e/Z_\delta)^2. \tag{31}$$

where Z_δ is $1-\frac{\delta}{2}$ percentile for standard Gaussian distribution and $\delta=1-\alpha$. Therefore, we first set f and l such that (31) is satisfied. Then we will decrease f and l empirically to minimize the execution time. The process is terminated until (31) is not satisfied and we pick the last pair (f,l) as the optimal value.

The snapshot size f and the virtual snapshot size l are computed from (31), which can be expanded from (29) and further expanded from (24), (26), (30) and other equations in Section III-D. In the end, the computation depends on the knowledge of the following parameters: the number k of tag sets, the number t_i of tags in the ith set, and the number n_i of tags from category cid in the ith set, where cid is an arbitrary category under consideration. Note that cid should appear as the superscript which we remove in the text for clarity as is stated earlier in the paper. The goal of the paper is to estimate the number n_* of common tags from category cid among all k tag sets, which cannot be derived from the above information. It may appear surprising that the estimation variance (29) does not depend on the number m of categories and the sizes of other categories. The reason is that we estimate n_* from category cid's virtual snapshot, which contains information of category cid and noise from other categories. The noise is dependent on the aggregate number of tags in other categories (related to t_i and n_i), regardless of which exact categories they come from.

In some application cases, we know the value of k. For example, suppose we want to study the volume of common tags (products) through a supply chain that comprises k distribution facilities of concern. The value of k is known. However, if we want to study the common tags among an arbitrary selection of some tag sets from a distribution network, we have to set k as the number of tag sets on the longest chain of interest in the network.

Unfortunately, the values of t_i and n_i are not pre-known. They need to be substituted with empirical upper bounds based on the past measurements. There exist a large number of efficient protocols [27], [30], [35], [36], [41] that can be used to measure the total number t_i of tags and the number n_i of tags in each category at a certain location, with a time complexity as low as $\Theta(\frac{1}{\epsilon^2}+loglog(t_i))$. Taking the maximum values of t_i and n_i measured in the past over different tag sets and different categories, we use these empirical upper bounds in our computation for f and l, which guarantees the accuracy requirement (1). Using upper bounds to compute

parameters is common in the related RFID literature [1]–[3], [5], [6], [8], [26]–[30], [35], [36], [41]. Since upper bounds are used, the actual accuracy will be better. It is certainly true that if the actual values of t_i and n_i breach the upper bounds, the requirement (1) may no longer hold, but we can still compute the actual variance from (29), allowing us to know how good the estimations are.

IV. JECM Over Unreliable Channels

So far, JECM assumes that the wireless channels between the RFID reader and tags are reliable, where no channel errors will be produced in the communication. However, it is common in practice that the communication between a reader and a tag suffers noise/channel fading/interference from the surroundings such as nearby objects, human movements and so on. As a result, an empty time slot may be corrupted and turn out to be a busy slot. In this case, when we take a snapshot of one tag set, the original empty slot which is supposed to be translated into a zero bit will be translated into a one bit, which will introduce estimation errors. Besides, noise may also make an impact on busy slots. In practice, the noise and the transmissions from tags may partially cancel each other if they happen to have opposite phases when they reach the reader. And it is extremely unlikely that they will cancel each other out exactly. So as long as the reader can still detect some energy (which may come from the noise), that would-be busy slot can still be correctly detected and translated to a one bit in our snapshot. Therefore, in this paper, we mainly focus on the impact of channel noise on empty slots.

Below, we will evaluate the impact of channel errors under two different models: random error model and burst error model.

A. JECM Under Random Error Model

In the random error model, the impact of channel error is characterized by a parameter called error rate P_{err}, which indicates the probability for each slot to be corrupted by the channel noise. For example, if $P_{err} = 5\%$, a would-be empty slot has a chance of 5% to be turned into a busy one due to the channel noise.

We call JECM under the random error model as JECM-rem. Since each empty slot independently has a probability P_{err} to be turned into a busy slot, at the reader's side, the expectation of the fraction of zero bits in the snapshot it takes will be

$$E_{rem}(U) = (1 - \frac{1}{f})^t \cdot (1 - P_{err}), \tag{32}$$

The fraction of zero bits in the virtual bitmap will be

$$E_{rem}(V) = (1 - \frac{1}{l})^n (1 - \frac{1}{f})^{t-n} \cdot (1 - P_{err}). \tag{33}$$

Similarly, for combined bitmaps and virtual bitmaps, we have

$$E_{rem}(U_{c_1c_2\ldots c_i}) = (1 - \frac{1}{f})^{t_{c_1c_2\cdots c_i}} \cdot (1 - P_{err})^i, \tag{34}$$

$$E_{rem}(V_{c_1c_2\ldots c_i}) = (1-\frac{1}{l})^{n_{c_1c_2\cdots c_i}} \cdot (1-\frac{1}{f})^{t_{c_1c_2\cdots c_i} - n_{c_1c_2\cdots c_i}} \cdot (1 - P_{err})^i. \tag{35}$$

With $E_{rem}(U)$, $E_{rem}(V)$, $E_{rem}(U_{c_1\ c_2\ldots c_i})$, $E_{rem}(V_{c_1\ c_2\ldots c_i})$, we can analyze JECM-rem similarly to JECM as in Section III if we replace (10), (11), (15) and (19) with (32), (33), (34) and (35), respectively.

B. JECM Under Burst Error Model

We now consider the burst error model. According to [42], the number of busts and the number of errors in each burst can be approximated with Poisson distribution. The probability density function (PDF) for the number of bursts is given by

$$h(x) = \sum_{i=0}^{\infty} \frac{\eta^i}{i!} e^{-\eta} \delta(x - i). \tag{36}$$

where η is the average number of bursts and $\delta(\cdot)$ is the Dirac Delta Function [43]. Meanwhile, according to convolutional codes and trellis code modulations, the PDF for the number of errors in a burst is given by

$$g(y) = \sum_{w=1}^{\infty} P_E(w) \delta(y - w). \tag{37}$$

where $P_E(w)$ represents the probability of having w errors in a burst

$$\begin{aligned} P_E(w) &= P(w - 1 < z_c \le w) = \int_{w-1}^{w} \sum g_c(z) dz \\ &= e^{-2\mu w}[(e^{2\mu} - 1)(1 + 2\mu w) - 2\mu e^{2\mu}]. \end{aligned} \tag{38}$$

$g_c(z)$ is the Erlang distribution of second order

$$g_c(z) = (2\mu)^2 z e^{-2\mu z}. \tag{39}$$

The probability of having w errors in an interval of f bits is

$$P_f(w) = \begin{cases} P_B(0), & \text{when } w = 0, \\ \sum_{j=1}^{\infty} P_e^{(j)}(w) P_B(j), & \text{when } w > 0. \end{cases} \tag{40}$$

where $P_e^{(j)}(w)$ is the probability to have w errors in j bursts in the interval of f bits

$$P_e^{(j)}(w)) = \begin{cases} P_E(w), & \text{when } j = 1, \\ \sum_{n=1}^{w} P_e^{(j-1)}(w - n) P_E(n), & \text{when } j > 1. \end{cases} \tag{41}$$

and $P_B(j)$ is the probability of having j bursts in an interval

$$P_B(j) = \frac{\eta^j}{j!} e^{-\eta}. \tag{42}$$

From (38), (40), (41) and (42), we know that the computation of $P_f(w)$ relies on η and μ. According to [42], the value of η can be computed based on the probability that a burst occurs and causes errors in the interval of f bits

$$\eta = \frac{f\beta}{N_e}. \tag{43}$$

where N_e is the mean value of the distribution $g(y)$ and can be computed as

$$N_e = E\{g(y)\} = \frac{e^{2\mu}(e^{2\mu} + 2\mu - 1)}{(e^{2\mu} - 1)^2}. \tag{44}$$

and β is a parameter called *bit error rate*. Besides, N_e can be also computed as

$$N_e = \frac{f N_m}{p_1(f + L_m - 1)}. \tag{45}$$

where N_m is the mean value of errors per burst, L_m is the mean value of burst error length and p_1 is the probability of

having at least one error in the interval when a burst occurs. p_1 is given by

$$p_1 = 1 - (1 - \frac{N_m}{f + L_m - 1})^f. \quad (46)$$

From (43), (44) and (45), we learn that the computation of μ and η relies on N_m, L_m, β and f, which are determined by our system. For example, if we set $N_m = 9.5$, $L_m = 33.5$, $\beta = 10^{-3}$, and $f = 10$, η and μ can be computed respectively as 0.0041 and 0.52. Then $P_f(w)$ can be derived from (40).

After obtaining $P_f(w)$, we learn that each original zero bit has a probability $\frac{w}{f}$ to be corrupted and turned into a one bit when there are w errors caused by the burst noise in an interval of f bits. Therefore, the probability p_0 for each zero bit in our snapshot not to be corrupted by the burst noise is

$$p_0 = \sum_{w=0}^{f} P_f(w)(1 - \frac{w}{f}). \quad (47)$$

We call JECM under the burst error model as JECM-bem. Similar to JECM-rem, we have

$$E_{bem}(U) = (1 - \frac{1}{f})^t \cdot p_0, \quad (48)$$

$$E_{bem}(V) = (1 - \frac{1}{l})^n (1 - \frac{1}{f})^{t-n} \cdot p_0, \quad (49)$$

$$E_{bem}(U_{c_1 c_2 \ldots c_i}) = (1 - \frac{1}{f})^{t_{c_1 c_2 \ldots c_i}} \cdot p_0^i, \quad (50)$$

$$E_{bem}(V_{c_1 c_2 \ldots c_i}) = (1 - \frac{1}{l})^{n_{c_1 c_2 \ldots c_i}} \cdot (1 - \frac{1}{f})^{t_{c_1 c_2 \ldots c_i} - n_{c_1 c_2 \ldots c_i}} \cdot p_0^i. \quad (51)$$

The performance of JECM-bem can be analyzed as described in Section III by replacing (10), (11), (15) and (19) with (48), (49), (50) and (51), respectively.

C. Discussion on Unreliable Channels

In practice, we may measure the error rate by transmitting a test frame to see how many bits are wrong. We can then plug the measured values of P_{err} into the formulas to compute the measurement variance. We may also measure the error rate over time to find an upper bound, and use the upper bound in computing the system parameters f and l from updated formulas in Section IV, which will ensure the accuracy requirement (1) is met in unreliable channel, subject to the validity of the error upper bound.

Following [44], an alternative design is to embed error-testing bits (slots) in the actual frames for joint cardinality estimation. Because the value of P_{err} is needed to determine the optimal frame size for a given accuracy requirement under our absolute error model, this approach will trade non-optimal frame size for more accurate error rate measurement.

V. Implementation of a Bimap Collection Protocol

The proposed solution to the problem of anonymous category-level joint estimation requires each reader to take snapshots of its local tag set in the form of bitmaps. Unfortunately, we cannot implement our work entirely by the commercial EPC C1G2 tags. For example, our protocol requires each tag to receive multiple seeds and use them in determining which bit to encode its presence. The circuit of today's C1G2 tags does not do that. Nonetheless, we show that the EPC C1G2 protocol [45] can be reconfigured to support bitmap collection, which means that with the enhancement of multi-seed reception and the computation of slot index $H_{h(tid)}(cid)$ for transmission (Section III-B), the C1G2 tags could be augmented to support the proposed work. As an example, Tash [20] uses a 128-bit memory bank available on commercial tags to produce 128 hash values, each of which can be used as a slot index and serve implicitly the purpose of a seed under our design. Hence, it can support our protocol in principle but put an upper limit of 128 seeds, which is too small. In contrast, the number of seeds used in our simulations is over 1000. If the future tags expand this memory bank to thousands of bits, the Tash method will support our protocol implementation, with an added benefit of removing the need for actually storing the seeds thanks to its memory bank based design.

The C1G2 protocol is originally designed for a reader to collect tag IDs. Below we show how to reconfigure it for collect a bitmap that encodes a tag set. We want to point out that many prior work on other RFID functions [1]–[3], [5], [6], [8], [26]–[31], [35], [36], [41] can also benefit this bitmap collection protocol.

A. EPC C1G2 ID Collection Protocol

The EPC C1G2 protocol [45] specifies the physical and logical requirements for a passive-backscatter RFID system that operates in 860MHz $\sim$ 960MHz. It is supported by most commercial passive RFID tags. The protocol collects the IDs of all tags within the reader's radio coverage.

The reader initiates an inventory round by broadcasting a `Query` command, which is 22 bits long and includes a parameter Q. The `Query` command is followed by an ALOHA frame, which consists of 2^Q time slots, in which tags can transmit responses. Upon receiving a `Query`, each tag will choose a random value r in the range $[0, 2^Q - 1]$ and load r into its slot counter. The counter is reduced by one for each slot; the reader starts each slot except for the first one with a `QueryRep` broadcast. If the counter of a tag is greater than zero, the tag will not transmit in the slot. But when the counter is reduced to zero (or it is zero initially), the tag will send a short response to the reader right after `QueryRep` (or `Query` if it is the first slot). The short response includes a 6-bit preamble, a 16-bit random number `RN16`, and a dummy bit. The reader listens to tag response in each slot. There are three cases:

- Case 1. When a single tag responds in a slot, the reader will resolve the response for `RN16` and it will broadcast an `ACK` command, containing the `RN16`.
- Case 2. When no tag responds, the reader will broadcast an `ACK` command, without a 16-bit resolved number. No tag will transmit an ID, and the reader will transmit a 4-bit `QueryRep` command to start the next slot.
- Case 3. If multiple tags choose the same random value to load into their slot counters, they will respond in the same slot, causing collision. In this case, the reader will receive mixed signals of multiple random `RN16`s and resolve into an `RN16` that is unlikely to match any of the original ones.

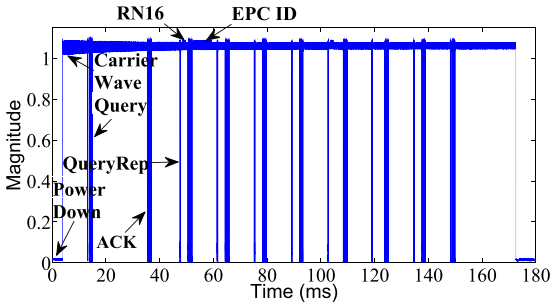


Fig. 3. Communication between a reader and tags when collecting tag IDs.

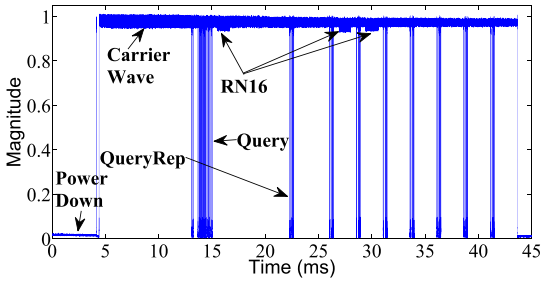


Fig. 4. Communication between a reader and tags when collecting a bitmap.

When a tag receives the `ACK` command, it will resolve the `RN16` carried by the command. If the resolved `RN16` matches its own `RN16`, the tag will transmit a tag-ID response, which includes a 6-bit preamble, a 16-bit Protocol Control (PC) word, a 96-bit EPC ID word, a 16-bit CRC-16 and a dummy bit.

The reader keeps listening to the channel. If it receives a tag-ID response, it will resolve the ID. It may also happen that no tag transmits ID. In either case, the reader will transmit a 4-bit `QueryRep` command to start the next slot.

After receiving the `QueryRep` command, each tag decreases its slot counter by one and will respond to the reader in the slot when the counter is decreased to zero. An inventory round is over when all 2^Q slots are broadcast by the reader. Multiple rounds may be needed to collect the IDs of all tags.

B. Bitmap Collection Protocol

In our solution to the problem of anonymous category-level joint estimation, each tag will transmit one bit information, instead of its ID. We show that the standard EPC C1G2 protocol can be reconfigured to serve this purpose, where each slot will deliver one bit information to the reader and the slot size can be made much smaller.

As is stated in Section II-A, the status of each slot is classified into two types: *empty slot* and *busy slot*. Following the EPC C1G2 protocol, the reader initiates communication by broadcasting a `Query` command, which is followed by an ALOHA frame. In each slot, the reader either receives one or multiple mixed short tag responses (i.e., `RN16`) or observes an idle channel — in the former case, this is a busy slot for a bit "1"; in the latter case, this is an empty slot for a bit "0". After that, instead of transmitting `ACK`, we reconfigure the reader to broadcast `QueryRep`, which cuts the slot short and starts the next slot. Since the reader never broadcasts an `ACK` command, no tag-ID response will be transmitted by any tag. The protocol will execute a single ALOHA frame, which will be converted into a bitmap: "1" for every busy slot and "0" for every empty slot.

In comparison, each slot in the original ID collection protocol contains 23 + 135 bits, whereas each slot in the bitmap collection protocol contains only 23 bits.

We implement a UHF (ultra high frequency) RFID reader on the USRP (universal software radio peripheral) platform, following [46]. We use two Laird antennas [47], one as a transmitter and the other as a receiver. The signals of one experimental run of the original EPC C1G2 protocol [45] with three commercial Alien Squiggle UHF RFID tags [48] are shown in Fig. 3, where the horizontal axis is time in ms and the vertical axis is signal magnitude. Initially, only noise exists during Power Down. After powering up, Carrier Wave is shown near the top of the figure. When the reader transmits (e.g., `Query` and `QueryRep`), signal magnitude changes greatly, shown as vertical lines in the figure. When a tag responds by backscattering, the carrier wave (top bold line in the figure) is modulated slightly. We set $Q = 3$ so the frame consists of 8 slots.

The signals of another experimental run for the reconfigured protocol of bitmap collection is shown Fig. 4, where the first, third and fourth slot are busy slot, with short responses being received by the reader after the `Query/QueryRep` command, while the other five slots are empty. As a result, this frame is converted to a bitmap of "10110000", which is a snapshot of the tag set.

VI. Simulation Results

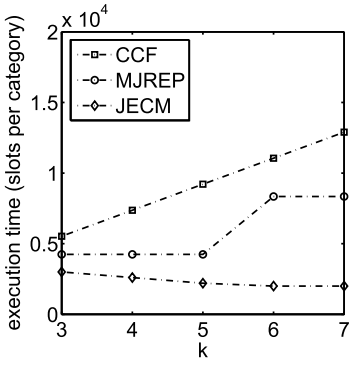
The proposed protocols are designed for large RFID systems with tens or even hundreds of thousands of tags where protocol efficiency becomes critical. For large-scale evaluation with numerous categories and tags, we resort to simulations.

A. Performance Evaluation Under Reliable Channels

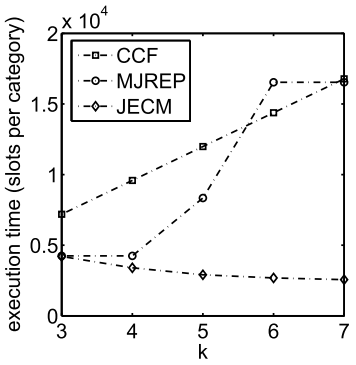
We first evaluate by simulations the performance of multi-set category-level joint estimation protocols under reliable channels. There is no prior work on estimating category-level joint information over an arbitrary number of tag sets. The most related work to our problem is CCF [35] and MJREP [36], but their protocols were designed to perform cardinality estimations over multiple sets but not at category level. As discussed earlier, we can adapt CCF and MJREP to perform estimation on one category at a time: The reader picks a category ID cid from M to broadcast in a request. A tag will participate in the execution of CCF (or MJREP) if and only if its category ID matches cid. In this way, we can repeat the protocol to estimate the cardinality of one category at a time. Although this adaptation loses anonymity, we can still use these protocols for comparative evaluation in terms of time efficiency.

We use the performance metrics in Subsection II-C for evaluation. We will first compare the execution times of JECM, CCF and MJREP, subject to the same accuracy requirement under perfect, reliable channels. Then, we will evaluate how well the proposed JECM can achieve a given accuracy requirement. Finally, we will investigate and compare the anonymity of these three protocols.

The system model is a distributed RFID system of k locations, with an accuracy requirement of $e = 50$ and $\alpha = 90\%$ or 95%. At each location, a reader periodically takes a snapshot



(a) $\alpha = 90\%, e = 50$



(b) $\alpha = 95\%, e = 50$

Fig. 5. Execution time comparison with respect to number of tag sets, subject to the same accuracy requirements.

of the local tag set. We set the number m of categories in each set to be 500 and the number of tags in each category to be 1000. We let the number n_* of common tags follow a zipf distribution [49] in $[10, 1000]$ and vary k from 3 to 7. We set $l_{cid} = 16$ out of $l_{id} = 96$. We will vary the values of m and n in our simulations in the next subsection, which includes comparison of the protocol performance in reliable and unreliable channels of different error rates.

We set the parameters for JECM based on Section III-F. And the parameters of CCF and MJREP are set by exactly following [35], [36]. Specifically, for CCF, the length value is $\lceil \log(k*1000) \rceil$ and the number of synopses is $\Theta(\frac{1}{\varepsilon^2 J} \ln \frac{1}{1-\alpha})$; for MJREP, f is optimized as is described in [36].

1) Execution Time: The first set of simulations evaluate the average protocol execution time under different accuracy requirements. Fig. 5 compares their execution times when $\alpha = 90\%, e = 50$ and $\alpha = 95\%, e = 50$. In each plot, the x-axis is the number k of tag sets, and the y-axis is the average number of slots needed per category by each protocol. Comparing Fig. 5a with 5b, we can observe that when the accuracy requirement is higher, the execution time needed is longer. This is expected since the reader needs to allocate a larger frame to increase the estimation accuracy, resulting in a longer execution time. When $k = 3$, all three protocols have comparable time costs, while CCF and MJREP takes longer than JECM. As k increases, the execution time of JECM decreases, while CCF and MJREP take longer execution time. The reason is that: For JECM, a larger number of tag sets provide more opportunity to filter out non-common tags during the inclusion/exclusion set joint process, which means a smaller time frame can be used to meet a certain accuracy requirement, resulting in smaller execution time. For CCF and MJREP, by doing one category at a time, the small number of common tags will take a larger time frame to separate them out from other tags, which is not a problem for JECM that records all categories together, ensuring a larger number of common tags. The curve of MJREP takes the non-smooth shape because the time frame for each category is set to a power of 2, with a large discrete jump between different settings. For a specific comparison, for joint estimation over 5 tag sets when $\alpha = 0.95, e = 50$, JECM needs 2,902 slots per category, while CCF and MJREP need 11,983 and 8,192 slots respectively.

2) Estimation Accuracy: The second set of simulations evaluate the accuracy of JECM. We vary the number k of tag sets from 3 to 7 and set the system parameters based on the description above. Fig. 6 shows the results from joint estimation over 3, 4 and 5 tag sets of 500 categories under the accuracy requirement of $\alpha = 95\%, e = 50$. Each point in the plot represents one category, where the x coordinate is the number n_* of common tags and the y coordinate is the estimated value $\hat{n}_*$. The equality line, $y = x$, is drawn for reference: the closer a point is to the equality line, the more accurate the estimation result is. From this figure, we can observe that most estimation results are clustered around the equality line, demonstrating good accuracy of our protocol under different numbers of tag sets. Fig. 7 shows the cumulative distribution function (CDF) of estimation errors. The x coordinate is the estimation error, the y coordinate is the probability for the estimation error to fall below this range and the red dotted line is the error bound we set. For $k = 3$, 4 and 5, the probabilities for estimation error being bounded by 50 are 0.954, 0.952 and 0.964 respectively, which confirm that JECM can indeed meet the pre-defined accuracy requirement of $\alpha = 95\%$ and $e = 50$ in all simulation cases.

3) Anonymity: The third set of simulations investigate anonymity of JECM, CCF and MJREP. Recall that in Table II, $p_{id} \approx 1$ for all these three protocols when $l_{id} = 96$ and $l_{id} - l_{cid} = 80$. So we only study the p_{cid} of these the protocols.

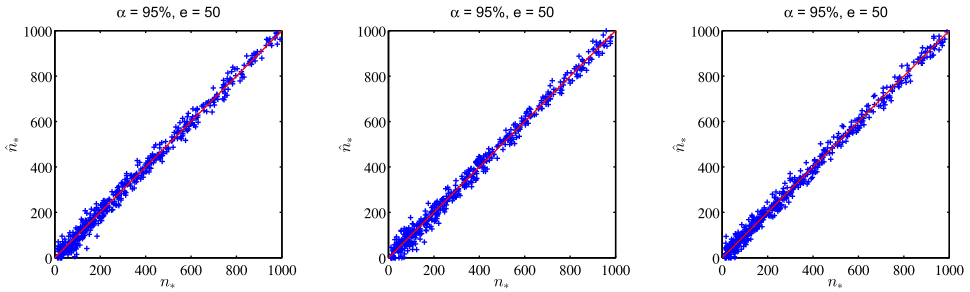
TABLE III
PRESERVED *cid* ANONYMITY OF JECM UNDER GIVEN SIMULATION SETTINGS

k	3	4	5	6	7
p_{cid}	98.90%	98.28%	98.99%	98.72%	99.02%

Table III shows the preserved anonymity of JECM when the number k of tag sets varies from 3 to 7. Each column represents the corresponding preserved anonymity p_{cid} of JECM when performing joint estimation on k tag sets. The table shows that the p_{cid} values are close to 1 in all simulations of JECM, which means the probability for an unauthorized adversary to reveal any category ID is very low. (The slight variance among the p_{cid} values is due to the randomness in simulations.) For CCM and MJREP, since the reader must broadcast category IDs one at a time before each round of estimation, any unauthorized adversary that eavesdrops the communication channel can easily acquire these IDs, making $p_{cid} = 0$ for both protocols.

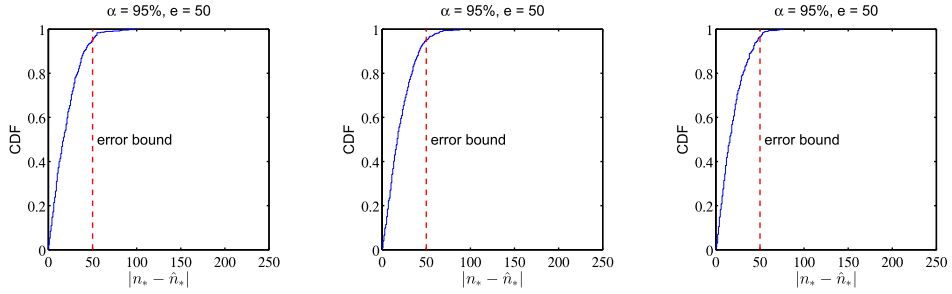
B. Performance Evaluation Under Unreliable Channels

In this section, we evaluate the performance of JECM-rem and JECM-bem. To simulate the random error model of error



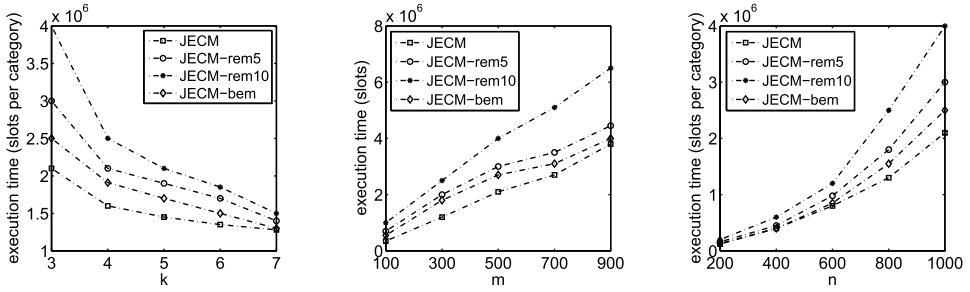
(a) $k = 3$ (b) $k = 4$ (c) $k = 5$

Fig. 6. Estimation results for $k = 3, 4, 5$ sets with 500 categories.



(a) $k = 3$ (b) $k = 4$ (c) $k = 5$

Fig. 7. CDF of estimation errors for $k = 3, 4, 5$ sets with 500 categories.



(a) Impact of k. (b) Impact of m. (c) Impact of n.

Fig. 8. Execution time comparison among different protocols with respect to different parameters.

rate P_{err} in JECM-rem, we use a pseudo-random number generator, which uniformly generates random real numbers in the range of $[0, 1]$. If a bit in the snapshot is "0" and the random number generated is within the range of $[0, P_{err}]$, we flip that bit to "1", which simulates one random error. To simulate the burst error model in JECM-bem, we pre-compute the values of $P_f(w)$ with different w for a given f. Then for each interval of f bits in our snapshot, we generate a random number in the range of $[0, 1]$ and check with $P_f(w)$ which w it corresponds, thereby determining the number of errors in the interval. For an interval that has w burst errors, we first generate a real random number in the range of $[0, 1]$. If a bit in this interval is "0" and the random number generated is within the range of $[0, \frac{w}{f}]$, we flip that bit to "1", which simulates one burst error.

We set $\alpha = 95\%$ and $e = 50$. For JECM-rem, we set $P_{err} = 5\%$ and 10% and call them JECM-rem5 and JECM-rem10 respectively. For JECM-bem, we follow [42] and set $N_m = 9.5$, $L_m = 33.5$, $\beta = 10^{-3}$, and $f = 100$. As a result, $\eta = 0.014$ and $\mu = 0.15$. Note that the protocol always satisfies the accuracy requirement specified by α and e, and our evaluation is to show how different channel errors will affect the protocol's time efficiency.

1) Number of Sets: The fourth set of simulations evaluate and compare the performance of JECM, JECM-rem5, JECM-rem10 and JECM-bem under different number k of tag sets. We set the number m of categories in each set to be 500 and the number n of tags in each category to be 1000. We let the number n_* of common tags follow a zipf distribution in $[10, 1000]$ and vary k from 3 to 7.

Fig. 8a compares their execution times. The x-axis is the number k of tag sets, and the y-axis is the number of slots needed by each protocol. It is not surprising that our protocol takes more time under unreliable channels. And as the noise level increases, the execution time needed also increases since we need a larger frame to tolerate the interferences from channel errors under the same estimation accuracy requirement. However, as k increases, the execution time of all these four protocols still decreases and the impact of channels errors is also degraded. On the one hand, a larger number of tag sets

provide more chances to filter out non-common tags during the inclusion/exclusion set joint process, which means a smaller time frame can be used to meet a certain accuracy requirement, resulting in smaller execution time. On the other hand, more tag sets also provide more chances to filter out channel errors during the inclusion/exclusion set joint process, which impair the impact of channel errors.

2) Number of Categories: The fifth set of simulations evaluate and compare the performance of JECM, JECM-rem5, JECM-rem10 and JECM-bem under different number m of categories. We set the number k of sets to be 3 and the number n of tags in each category to be 1000. We let the number n_* of common tags follow a zipf distribution in $[10, 1000]$ and vary m from 100 to 900 at a step size of 200.

Fig. 8b compares their execution times. The x-axis is the number m of categories in each tag set, and the y-axis is the number of slots needed by each protocol. Similarly, the channel errors introduced by the unreliable channels increase the execution time of our protocol. The higher the noise level is, the more time our protocol takes. Besides, it is also expected that it takes more time for each protocol to incorporate with more categories in each tag set. More categories in each tag set will produce more tags. To maintain the estimation accuracy, we need to allocate a larger frame to accommodate more tags, resulting in longer execution time.

3) Size of Each Category: The sixth set of simulations evaluate and compare the performance of JECM, JECM-rem5, JECM-rem10 and JECM-bem under different size n of each category. We set the number k of sets to be 3 and the number m of categories in each set to be 500. We vary the size n of each category from 200 to 1000 at a step size of 200 and let the number n_* of common tags follow a zipf distribution in $[10, n]$.

Fig. 8c compares their execution times. The x-axis is the size n of each category, and the y-axis is the number of slots needed by each protocol. Similarly, a higher noise level requires longer execution time since we need a larger frame to tolerate more noises. Besides, it is also expected that it takes more time for each protocol to incorporate with more tags in each category, since the reader needs to allocate a larger frame to maintain the estimation accuracy.

VII. Related Work

There is no prior work directly designed for anonymous category-level joint estimation over multiple sets. We discuss the various work on the related problems.

Measuring the number of tags in a system can be done by identifying the IDs of all tags. Existing tag identification protocols such as DFSA [50] collect all IDs using the EPC C1G2 standard [45]. One problem of this approach is that it is not time-efficient when there is a large number of tags and the operation must be frequently performed in a dynamic system. Due to collisions, the lower bound for the number of slots needed in tag identification is $e \times n \times 96$ [50], where e is natural constant, n is the number of tags to be identified, and 96 is the length of tag ID. Moreover, identifying tag IDs compromises anonymity.

Another direction that researchers have pursued is to estimate the cardinality of a tag set without identifying tag IDs. To minimize the time cost, a series of protocols have been proposed, including generalized maximum likelihood estimation [51], lottery frame protocol [52], PET [25], unified probabilistic estimator [1], zero-one estimator [28], etc. These protocols adopt the relative error model and can efficiently estimate the cardinality of one tag set without revealing tag IDs. However, these protocols are not feasible in our problem. First, the relative error model does not work well in our settings as is explained previously. Second, these protocols are designed for single-set cardinality estimation and cannot handle multiple sets, let alone at the category level.

Most related are DTE [33], CCF [35], JREP [34], MJREP [36] and ZDE [31]. As is mentioned earlier, DTE, JREP and ZDE are designed to estimate the joint information of two tag sets. And their solutions cannot be easily extended to an arbitrary number of tag sets. CCF is designed to estimate the cardinality of arbitrary set expression with desired accuracy. It exploits a synopsis for estimating each tag set and the size of the synopsis is sublinear to set cardinality. CCF adopts the relative error model in estimation and the time cost for joint estimation is $\Theta(\frac{1}{\varepsilon^2 J} \ln \frac{1}{1-\alpha})$, which approaches to infinity as n^* approaches to zero, where J is the Jaccard similarity, $J = \frac{n_*}{n}$ and n is the number of tags in all tag sets. MJREP is another protocol that estimates joint information of multiple sets. It takes two rounds for MJREP to perform the estimation. In the first round, MJREP estimates the cardinality of a tag set and adaptively sets a proper frame size. In the second round, the estimation is performed with the optimized frame size. However, both protocols are not designed for category-level joint estimation. We may adapt them for that purpose, but their efficiencies are inferior and they break anonymity, as we have evaluated and discussed earlier. Another related work is [38] which can only perform joint estimation on two tag sets. And the methodology it uses can not be easily extended to the joint estimation of more sets, which is required by many real-world applications.

VIII. Conclusion

This paper studies a new problem of anonymous category-level joint estimation over multiple tag sets in RFID systems: for any category in a large RFID system, we want to anonymously estimate the cardinality of the intersection among multiple tag sets. We design a protocol called JECM based on temporal or spatial snapshots. We derive an estimator, perform statistical analysis on it, and provide formulas for optimizing system parameters. Moreover, we extend JECM to work under unreliable channels. Through extensive simulations, we evaluate the performance of our protocol and demonstrate that our protocol outperforms the prior art in time cost reduction and anonymity preservation.

References

[1] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2006, pp. 322–333.

[2] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. 11th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, Sep. 2010, pp. 1–10.

[3] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[4] J. Han, Y. Zhao, Y. S. Cheng, T. L. Wong, and C. H. Wong, "Improving accuracy for 3D RFID localization," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 5, May 2012, Art. no. 865184.

[5] T. Li, S. Chen, and Y. Ling, "Efficient protocols for identifying the missing tags in a large RFID system," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1974–1987, Dec. 2013.

[6] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu, "Completely pinpointing the missing RFID tags in a time-efficient way," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 87–96, Jan. 2015.

[7] J. Ou, M. Li, and Y. Zheng, "Come and be served: Parallel decoding for COTS RFID tags," in *Proc. ACM MobiCom*, 2015, pp. 500–511.

[8] M. Shahzad and A. X. Liu, "Fast and reliable detection and identification of missing RFID tags in the wild," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3770–3784, Dec. 2016.

[9] L. Yang, Q. Lin, X. Li, T. Liu, and Y. Liu, "See through walls with COTS RFID system!" in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2015, pp. 487–499.

[10] L. Yang, Y. Li, Q. Lin, X.-Y. Li, and Y. Liu, "Making sense of mechanical vibration period with sub-millisecond accuracy using backscatter signals," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2016, pp. 16–28.

[11] X. Liu *et al.*, "Top-k queries for categorized RFID systems," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2587–2600, Oct. 2017.

[12] J. Liu, Y. Zhang, M. Chen, S. Chen, and L. Chen, "Collision-resistant communication model for stateless networked tags, poster paper," in *Proc. IEEE ICNP*, 2016, pp. 1–2.

[13] Y. Hou, J. Ou, Y. Zheng, and M. Li, "PLACE: Physical layer cardinality estimation for large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2702–2714, Oct. 2016.

[14] G. Wang *et al.*, "HMRL: Relative localization of RFID tags with static devices," in *Proc. 14th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2017, pp. 1–9.

[15] J. Han *et al.*, "GenePrint: Generic and accurate physical-layer identification for UHF RFID tags," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 846–858, Apr. 2016.

[16] Y. Zhang, S. Chen, Y. Zhou, and Y. Fang, "Using wireless tags to monitor bodily oscillation," in *Proc. IEEE 15th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2018, pp. 211–219.

[17] H. Ding *et al.*, "RFIPad: Enabling cost-efficient and device-free in-air handwriting using passive tags," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 447–457.

[18] J. Liu, Y. Zhang, S. Chen, M. Chen, and L. Chen, "Collision-resistant communication model for state-free networked tags," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 656–665.

[19] C. Duan, X. Rao, L. Yang, and Y. Liu, "Fusing RFID and computer vision for fine-grained object tracking," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[20] L. Yang, Q. Lin, C. Duan, and Z. An, "Analog on-tag hashing: Towards selective reading as hash primitives in Gen2 RFID systems," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 301–314.

[21] Y. Zhang, S. Chen, Y. Zhou, and Y. Fang, "Anonymous temporal-spatial joint estimation at category level over multiple tag sets," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 846–854.

[22] Y. Zhang, S. Chen, Y. Zhou, and O. Odegbile, "Missing-tag detection with presence of unknown tags," in *Proc. 15th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2018, pp. 1–9.

[23] M. Chen, S. Chen, Y. Zhou, and Y. Zhang, "Identifying state-free networked tags," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1607–1620, Jun. 2017.

[24] C. C. Tan, B. Sheng, and Q. Li, "How to monitor for missing RFID tags," in *Proc. 28th Int. Conf. Distrib. Comput. Syst.*, Jun. 2008, pp. 295–302.

[25] Y. Zheng, M. Li, and C. Qian, "PET: Probabilistic estimating tree for large-scale RFID estimation," in *Proc. 31st Int. Conf. Distrib. Comput. Syst.*, Jun. 2011, pp. 37–46.

[26] T. Li, S. Wu, S. Chen, and M. Yang, "Energy efficient algorithms for the RFID estimation problem," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[27] M. Shahzad and A. X. Liu, "Every bit counts: Fast and scalable RFID estimation," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. (Mobicom)*, 2012, pp. 365–376.

[28] Y. Zheng and M. Li, "Towards more efficient cardinality estimation for large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1886–1896, Dec. 2014.

[29] W. Luo, Y. Qiao, and S. Chen, "An efficient protocol for RFID multigroup threshold-based classification," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 890–898.

[30] L. Xie, H. Han, Q. Li, J. Wu, and S. Lu, "Efficiently collecting histograms over RFID tags," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2014, pp. 145–153.

[31] M. Shahzad and A. X. Liu, "Fast and accurate tracking of population dynamics in RFID systems," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 836–846.

[32] W. Luo, Y. Qiao, S. Chen, and M. Chen, "An efficient protocol for RFID multigroup threshold-based classification based on sampling and logical bitmap," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 397–407, Feb. 2016.

[33] Q. Xiao, B. Xiao, and S. Chen, "Differential estimation in dynamic RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 295–299.

[34] Q. Xiao, M. Chen, S. Chen, and Y. Zhou, "Temporally or spatially dispersed joint RFID estimation using snapshots of variable lengths," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2015, pp. 247–256.

[35] H. Liu, W. Gong, L. Chen, W. He, K. Liu, and Y. Liu, "Generic composite counting in RFID systems," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst.*, Jun. 2014, pp. 597–606.

[36] Q. Xiao, S. Chen, and M. Chen, "Joint property estimation for multiple RFID tag sets using snapshots of variable lengths," in *Proc. 17th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2016, pp. 151–160.

[37] B. Sheng, C. C. Tan, Q. Li, and W. Mao, "Finding popular categories for RFID tags," in *Proc. 9th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, May 2008, pp. 159–168.

[38] M. Chen, J. Liu, S. Chen, and Q. Xiao, "Efficient anonymous category-level joint tag estimation," in *Proc. IEEE 24th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2016, pp. 1–10.

[39] K.-Y. Whang, B. T. Vander-Zanden, and H. M. Taylor, "A linear-time probabilistic counting algorithm for database applications," *ACM Trans. Database Syst.*, vol. 15, no. 2, pp. 208–229, Jun. 1990.

[40] M. Yoon, T. Li, S. Chen, and J. Peir, "Fit a compact spread estimator in small high-speed memory," *IEEE Trans. Netw.*, vol. 19, no. 5, pp. 1253–1264, Oct. 2011.

[41] B. Chen, Z. Zhou, and H. Yu, "Understanding RFID counting protocols," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2013, pp. 291–302.

[42] B. Cornaglia and M. Spini, "Letter: New statistical model for burst error distribution," *Eur. Trans. Telecommun.*, vol. 7, no. 3, pp. 267–272, May 1996.

[43] (2017). *Dirac Delta Function*. [Online]. Available: http://en.wikipedia.org/wiki/Dirac_delta_function

[44] Z. Zhou and B. Chen, "RFID counting over time-varying channels," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1142–1150.

[45] *Radio-Frequency Identity Protocols Class-1 Gen-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz, EPCglobal*. Accessed: Sep. 2011. [Online]. Available: http://www.epcglobalinc.org/uhfclg2

[46] M. Buettner and D. Wetherall, "A software radio-based UHF RFID reader for PHY/MAC experimentation," in *Proc. IEEE Int. Conf. RFID*, Apr. 2011, pp. 134–141.

[47] *Laird*. Accessed: May 2015. [Online]. Available: http://www.lairdtech.com/products/s9028PCL

[48] *AlienTags*. Accessed: Aug. 2014. [Online]. Available: http://www.alientechnology.com/products/tags/squiggle/

[49] *Zipf's Law*. Accessed: Dec. 2003. [Online]. Available: https://en.wikipedia.org/wiki/Zipf's_law

[50] C. T. Nguyen, K. Hayashi, M. Kaneko, P. Popovski, and H. Sakai, "Probabilistic dynamic framed slotted ALOHA for RFID tag identification," *Wireless Pers. Commun.*, vol. 71, no. 4, pp. 2947–2963, Aug. 2013.

[51] Z. Zhou, B. Chen, and H. Yu, "Understanding RFID counting protocols," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 312–327, Feb. 2016.

[52] C. Qian, H. Ngan, Y. Liu, and L. M. Ni, "Cardinality estimation for large-scale RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1441–1454, Sep. 2011.