IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON INTELLIGENT SYSTEMS FOR THE INTERNET OF THINGS

Received November 9, 2017, accepted December 11, 2017, date of publication December 27, 2017,
date of current version March 9, 2018.

*Digital Object Identifier 10.1109/ACCESS.2017.2785810*

# Fast Discrepancy Identification for RFID-Enabled IoT Networks

**CAIDONG GU**
Department of Computer Science, Suzhou Vocational University, Suzhou 215104, China
e-mail: gucaidong6688@163.com

This work was supported by NSFC under Grant 61472268.

**ABSTRACT** Radio frequency identification has become a vital technique for enabling intelligent supply chain management. Throughout the life cycle of the supply chain, one-for-one checking is necessary at all the handover points where physical shipped inventory is checked against with the receiver's order to discover discrepancies because of missing or misplaced objects. Such operation is so universal that the improvement over its efficiency can significantly benefit the whole supply chain. Yet, there are a few challenging issues, namely, inconsistent tags, high-volume data, and high network latency. In this paper, we first carefully analyze the characteristics of EPCglobal Network. We then design two discrepant tag identification protocols with different optimization goals including minimum communication data and minimum communication round. Also, we perform thorough analysis by comparing our proposals with state-of-the-art methods and extensive experiments to validate the effectiveness and efficiency of our methods.

**INDEX TERMS** RFID, IoT, tag identification.

## I. INTRODUCTION

Radio Frequency Identification (RFID) has become a sweeping technology in IoT-enabled supply chain management all over the world [1]–[3]. A typical supply chain consists of four processes tracking the goods flow: from supplier to manufacturer, to distributor, to retailer, and to end-customer. Formerly, supply chain management is often a labor consuming process. Yet after the adoption of RFID technologies, the efficiency of supply chain management has been greatly improved. For instance, Walmart saved nearly 300 million dollars from RFID-enabled inventory [4]. There are three major advantages of applying RFID for supply chain management. First, the efficiency of manufacturing could be largely improved. This is manly because thousands of RFID tags can be read at the same time whereas traditional barcode technologies can only read one object per time. Second, the whole life cycle of the supply chain can be effectively tracked, providing valuable visibility from manufacturing to shipping. Third, the delivery and dispatch speeds are significantly benefited from automatic registries and storage.

As demonstrated in Figure 1, in a supply chain, the end-customers buy stuff from some retailer. Accordingly, the retailer needs to get goods from distributors. Then the distributer shall obtain a large number of goods from manufactures, who get raw materials from the suppliers.

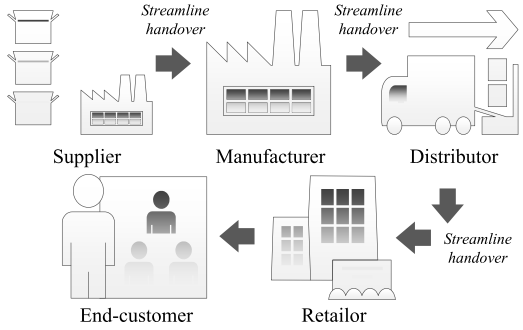


**FIGURE 1.** An example of retail supply chain.

At each handover point, one-for-one checking is necessary that the receiver should compare all the items actually obtained against the ship list. For example, when a bunch of goods are arriving at some distributor, the retailer shall conduct scanning to collect all the information from arrival goods from RFID-enabled pallets. Those information then are compared against the list from the original distributor. The comparison is supposed to find any discrepancies and fix them if possible. From this example, we can see that such checking operations are so common that its efficiency is the core of the whole supply chain management.

However, there are a number of challenges in optimizing one-for-one checking efficiency.

*Inconsistent Information:* There are several reasons for the mismatches between the actual ship list and the receiver's order, e.g., vendor fraud, human errors, and theft [5]. Even worse, the RFID system errors, including missing, counterfeit, and misplaced tags, are making discrepancy identification more challenging.

*Big Data:* In RFID-enabled supply chain, tons of RFID tags and readers have been deployed across every link of supply chain management. Those readers and tags are generating voluminous data at a very high speed. For example, there is nearly 7 TB data acquired everyday at Walmart [6]. So, a big issue is how to perform real-time analysis from those big RFID data from the supply chain.

*High-Latency Network:* Each scanning generates loads of data, including Electronic Product Code (EPC), arrival time, and location. All such data shall later be retrieved in the centralized EPCglobal Network, increasing network delay. Meanwhile, online operations create notable network latency because of the long range traffic. For example, 65% of the queries in EPCglobal network take more than 0.5s to respond even ideal network conditions are assumed [7].

There are two problems closed related to the discrepant tag identification: missing-tag identification and unknown-tag identification. Yet, missing-tag identification can only find the tags that are absent in a known list [8], [9] and unknown-tag identification can only discover the tags that are not included in a pre-defined list [10], [11]. In practical applications, e.g., one-on-one checking, the administrator needs to find the discrepant tags including both missing tags and unknown tags at the same time. We defer the detailed problem comparisons to the next section. To address this problem and aforementioned challenges, we propose efficient schemes for discrepant tag identification in large RFID-enabled supply chains. The major contributions are as follows.

1) We identify the characteristics of EPCglobal Network and inefficiencies of traditional methods and then propose two efficient discrepant tag protocols using characteristic polynomial, namely, probabilistic identification protocol with optimized communication (PIP-OC) and probabilistic identification protocol with optimized rounds (PIP-OR).
2) We analytically compare the proposed schemes with previous methods and discuss their relationships and usages.
3) We perform extensive experiments to verify the effectiveness of our proposals and compare them with state-of-the-art solutions. The results show that PIP-OC and PIP-PIP-OR outperform previous solutions in latency-sensitive networks.

## II. RELATED WORK

There are several topics that are closely related to this paper, including, missing-tag, tag population estimation, and batch authentication.

In the past few years, numerous RFID counting methods have been proposed. Most of them are optimized for communication time and reliability. In a seminar work, the Unified Probabilistic Estimator (UPE) is introduced based on the framed ALOHA model [12]. Then Zheng *et al.* [13] improve the estimation time by designing a estimation tree $\mathcal{O}(\log \log n)$. Meanwhile, Shahzad and Liu [14] use Average Run based Tag Estimation (ART) that uses average run length as the core to achieve 7x faster than the state-of-the-art solutions. To handle the arbitrary tag distributions, $A^3$ has been proposed using universal hashing for cardinality estimation [15]. Also, energy-efficient cardinality estimation scheme has been introduced using maximum likelihood [16]. A churn estimation scheme is proposed to provide quick estimations on the number of new tags in a monitored region [17]. A good survey along this line can be found at [18].

Another problem that is closely related is missing-tag. A bunch of missing-tag identification protocols are proposed in [19]. Another work that considers both the trusted and untrusted readers is introduced to solve the probing of identifying missing tags under untrust cases [20]. Later Luo *et al.* [21] try to examine the energy-efficient aspect in detail and propose a energy-time tradeoff for detecting missing tags. A new protocol, RFID monitoring protocol with unexpected tags (RUN), is proposed to detect and identify missing tags with required reliability when unexpected tags are included [9].

A great deal of authentication schemes, also known as unknown-tag problems, have been designed for RFID. For example, Physically Unclonable Function (PUF) [22] is introduced to verify the genuineness of a single tag that is possibly cloned by the tagID. Later several hash based [23] and novel tree structure are proposed to fast authenticate tags [24]–[26]. Then counterfeit estimation is introduce to give accurate approximated count of illegal tags [27]. Those schemes have heavy computation overhead, which is suitable when the number of counterfeit tags is large.

To compare discrepant tag identification with all the above schemes, we make a summary in Table 1. For the second row of Table 1, the user case is that both the physical list and ship list have the same quantities. In this case, if we simply use the same quantity of counterfeit tags to replace genuine tags, the counting scheme would fail to find anything abnormal. Therefore, only missing-tag and authentication schemes can find the discrepant tag while all the counting schemes fail. For the third row, the case becomes that the physical list contains a counterfeit tag. The missing-tag schemes cannot handle this case because no tag is missing while counting and authentication schemes can find there is an extra tag. For the fourth row, the situation is that the physical list has a missing-tag. All previous authentication schemes would fail in the case because all the tag in the physical list are genuine. Counting and missing-tag schemes can discover the discrepancy caused by such a missing event. From the above, we can see that none of counting, missing-tag, and authentication schemes can achieve discrepant tag identification in all of the

**TABLE 1. Illustrative comparison of estimation, missing-tag, authentication and discrepancies identification.**

| [1] | Counting | Missing-tag | Authentication | Discrepancies |
|---|---|---|---|---|
| S: ♠♠♠<br>P: ♠♠♡ | × | ✓ | ✓ | ✓ |
| S: ♠♠♠<br>P: ♠♠♠♡ | ✓ | × | ✓ | ✓ |
| S: ♠♠♠<br>P: ♠♠ | ✓ | ✓ | × | ✓ |

three cases. Therefore, a dedicated discrepancy identification scheme is much needed to find the unmatched tags.

In intelligent supply chain management areas, a efficient storage model that uses the bloom filter has been employed in [6]. At the same time, Lee and Chung [28] propose a novel encoding scheme that uses flow information of products in supply chains. Later Cao *et al.* [29] design a well-known distributed system that can process and track products simultaneously. But those solutions fail to take discrepant tags into consideration.

## III. PRELIMINARIES

### A. EPCglobal NETWORK

EPCglobal, a joint venture of EAN International and the Uniform Code Council (UCC), aims to foster global adoption and implementation of the EPCglobal Network across industries. EPCglobal Network starts with EPC, which is always embedded in a RFID tag to uniquely identify the object. In general, EPC is a bitstring that contains information about manufacturer, product ID (e.g., model), and particular object ID. When a reader read the tag, its EPC is transferred back to the reader. If any further information (e.g., manufacture date) are required about a tag/object, the reader needs to retrieve them from manufacturer through EPCglobal Network. Therefore EPCglobal Network can also be seen as a computer network that can share product data (mainly RFID data) between different supply chain partners.

The EPC Network is a set of network services that enable the sharing of RFID-related data throughout the supply chain. As shown in Figure 2, EPCglobal Network mainly comprises Object Naming Service (ONS), EPC Information Services (IS), and EPC Discovery Services (DS). ONS, which is highly similar to Domain Name System in Internet, is a service that takes EPC and return one or more matched URLs or IP-addresses. The ONS contains two layers. The first layer is Root ONS, which includes the directory of manufacturers who contribute information on EPCglobal Network. The second layer is Local ONS, which has the directory of products related to particular manufacturer. The EPC IS, which is a standard of EPC-related data sharing within and across network participants, provides network services, including storage and access to product information of matched EPC.

[1] P denotes physical list, S denotes ship list, ♠ is genuine tag, and ♡ is counterfeit tag. ✓and ×denote if the scheme can detect anything unusual.

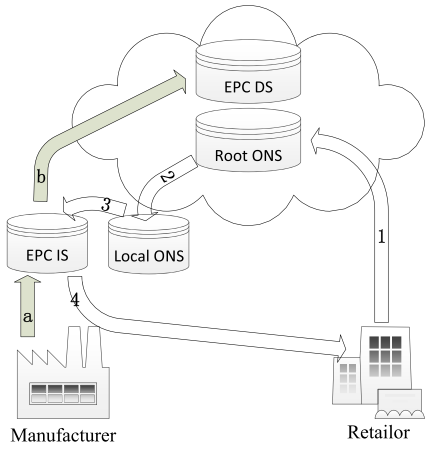


**FIGURE 2. An instance of EPCglobal network.**

The EPC DS, which is like search engine in Internet, is the registry of every EPC IS that contains information of a certain EPC/Object. Let's see two illustrative examples.

*Manufacture Process:* As showin in line a-b of Figure 2, when a manufacturer creates a product, its related tag information are recorded and stored with local EPC IS. Then the local EPC IS registers those information to global EPC DS for future reference.

*Information Retrieval Process:* As showin in line 1-4 of Figure 2, if a retailer requires further information of a certain product, the Root ONS is first queried to obtain the location of manufacturer's Local ONS. Then product related data is queried in corresponding EPC IS. Finally, the retailer gets desired information from manufacturer's EPC IS.

As reported in [7], EPCglobal networks do not operate very well in tracking because the network latency is high. Along with the movement of products in the supply chain, each site, e.g., manufacturer, distributor, has to record and retrieve some information from the products using RFID. Then, some EPC-IS instances need to register themselves with the EPC Discovery Service. This way, some network delays are incurred. In this paper, we aim to achieve tradeoffs between communication complexity and round complexity.

### B. PROBLEM FORMULATION

Here we formally formulate one-for-one checking problem. In this problem, there are two sides: sender and receiver. We mainly use two metrics: (i) communication complexity, which counts the total bits in the traffic; and (ii) round complexity, which is the number of single-way communication. Assume that each product/item has a composite data structure, TagData ($t$) of size $b_t$ bits. A TagData includes EPC and other tag info that are stored in the writable memory, such as: symetric/asymetric keys. Technically, the problem of discrepant tags identification is defined as follows: Provided a physical scanned list, denoted as $P = \{t_1^p, t_1^p, ..., t_n^p\}$ on the receiver and a ship list, denoted as $S = \{t_1^s, t_2^s, ..., t_m^s\}$ on the sender, the receiver needs to identify all the disagreed

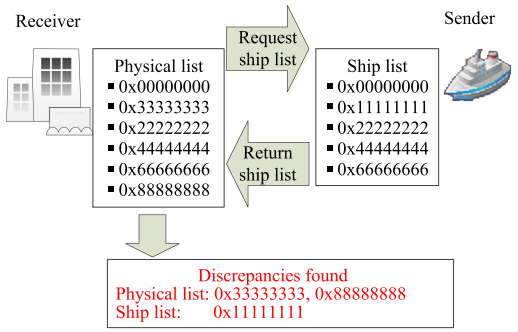


**FIGURE 3.** Traditional wholesale protocol.

tags by doing tradeoffs between communication and round complexity. For example, when keeping the round complexity at the minimal,1, what is the minimal communication complexity? Or while keeping the communication complexity at the minimal, what is the minimal round complexity?

### C. TRADITIONAL WHOLESALE PROTOCOL

The traditional Wholesale Protocol (WP) to solve discrepancies problem is shown in Figure 3. When receiver needs to verify the physical list on hand, a request is sent to sender. Then sender retrieves related RFID data and replies them in batch to receiver. Receiver compares ship list with its own physical scanned list to discover all discrepant tags. All above operations are done through EPCglobal Network and are not shown in the figure for simplicity.

Although WP is able to address discrepancies identification problem in simple setting, it suffers from inefficient communication in large-scale supply chain. The main reason is that information exchanged between receiver and sender in WP are proportional to the size of original ship list, but not the size of discrepant tags. Therefore, WP do not scale well with in practical scenarios: rapidly increasing volume of RFID data with relative a small amount of disagreed tags. In fact, due to online demands and continuous large volume of data in supply chain, discrepancies results are always required to be available in real-time. In short, efficient and reliable discrepant tags identification is very important for large-scale RFID-enabled supply chain.

As we will demonstrate in the followings sections, we design a series of efficient discrepancies identification protocols which incrementally improve the performance with balanced communication complexity and round complexity.

## IV. DETERMINISTIC IDENTIFICATION PROTOCOL

Our deterministic identification protocol (DIP) is built on characteristic polynomial [30], [31]. We can encode both the physical and ship lists using characteristic polynomials. In particular, instead of original raw data from RFID technologies, the sender transmits its own evaluated characteristic polynomial to the receiver. Then the receiver uses its own encoded data using characteristic polynomial to derive all the discrepant tags. For now, we assume that the number of discrepant tags is no more than $d$, which is a priori for both sides. We shall detail the encoding and decoding processes in the following.

*Encode:* According to the definition of characteristic polynomials, we can obtain the polynomials of a group of TagData, denoted as $T = \{t_1, t_2, ..., t_n\}$, as follows:

$$\mathcal{P}_T(t) = (t - t_1)(t - t_2)\cdots(t - t_n). \tag{1}$$

In addition, we use $\Delta P$ to denote the set of tags that are on the physical list but not on the ship list. Same way, we use $\Delta S$ to represent the set of tags that are on the ship list but not on the physical list. Let $Q$ be the set of tags that are on both lists. Then we can write the ratio of $P$ to $S$ using characteristic polynomials as follows,

$$\frac{\mathcal{P}_P(t)}{\mathcal{P}_S(t)} = \frac{\mathcal{P}_Q(t)\cdot\mathcal{P}_{\Delta P}(t)}{\mathcal{P}_Q(t)\cdot\mathcal{P}_{\Delta S}(t)} = \frac{\mathcal{P}_{\Delta P}(t)}{\mathcal{P}_{\Delta S}(t)}. \tag{2}$$

The above equation shows that the common elements on both lists cancel out each other by computing a ratio, which means $\Delta P$ and $\Delta S$ can be easily recovered by deriving the polynomial roots. Therefore, all the sender needs to do is to evaluate characteristic polynomial over $d$ sample points and transmits such polynomial values as codewords to the receiver because $d$ sample points are enough to recover the $\Delta P$ of which the degree is no more $d$. Note that it is easy to observe that if any one of the sample points is an element of $P$ or $S$, the ratio computation could fail. To solve this, we have a mandatory requirement that all arithmetic operations should be done on the finite field $\mathbb{F}_q$, ensuring $d$ sample points would not coincide with any elements, where $q \geq 2^{b_t} + d$.

*Decode:* Decoding polynomials of $P$ or $S$ could be daunting at first glance. Fortunately, we only need to focus on the $\mathcal{P}_{\Delta P}(t)$ and $\mathcal{P}_{\Delta S}(t)$ of which the degrees is much less than $P$ or $S$ thanks to the cancellation of all the common element in the ratio computation. At the same time, we know that a support set of size $d$ is enough to ensure the uniqueness of rational function based on the rational function interpolation in [32]. Hence, we just simply employ Gaussian elimination [33] to compute the desired rational functions with $d$ samples. Note that the associated time complexity for Gaussian elimination is only $\mathcal{O}(d^3)$.

Let's see a concrete example. Suppose we have physical scanned list $P = \{1, 2, 3, 5, 19\}$, ship list $S = \{2, 3, 5, 7, 8, 19\}$ stored as 6-bit strings, $d = 3$, and a sufficient large finite field $\mathbb{F}_{67}$. Thus we can have characteristic polynomial for $P$ and $S$ as:

$$\mathcal{P}_P(t) = (t - 1)(t - 2)(t - 3)(t - 5)(t - 19),$$
$$\mathcal{P}_S(t) = (t - 2)(t - 3)(t - 5)(t - 7)(t - 8)(t - 19).$$

Over $\mathbb{F}_{67}$, we evaluate characteristic polynomials on sample points $\{-1, -2, -3\}$ as follows.

| $t$ | -1 | -2 | -3 |
|---|---|---|---|
| $\mathcal{P}_P(t)$ | 1 | 24 | 52 |
| $\mathcal{P}_S(t)$ | 31 | 17 | 44 |
| $\mathcal{P}_P(t)/\mathcal{P}_S(t)$ | 13 | 29 | 56 |

Receiver applies rational function interpolation at sample points to obtain:

$$\frac{\mathcal{P}_{\Delta P}(t)}{\mathcal{P}_{\Delta S}(t)} = \frac{t+66}{t^2+52t+56}.$$

The zeros of numerator and denominator are $\{1\}$ and $\{7, 8\}$ that exactly match discrepancies of two lists.

*Analysis:* Because the size of finite field, $q$, is more than $2^{b_t} + d$, DIP needs one additional bit for each polynomial values, which is $b_t + 1$. Meanwhile, we have $d$ evaluated values to transmit. So the total bits for all the polynomials values are $(b_t + 1)d$. In addition, we have to let the receiver know the size of the list, which is of size $b_t$ bits. Hence the overall communication overhead is $(b_t + 1)d + b_t$ bits. Such communication overhead shows a nice feature of DIP, where the communication complexity grows linearly with the number of discrepant tags, $d$. Therefore, DIP is best fit for cases where the $d$ is relatively small, which is exactly the case for supply chain management.

## V. PROBABILISTIC IDENTIFICATION PROTOCOL WITH OPTIMIZED COMMUNICATION

In order to remove the requirement of knowing $d$ as a priori, we propose Probabilistic Identification Protocol with Optimized Communication (PIP-OC). The main idea of PIP-OC is to execute DIP with incrementally increased $d$ until it is large enough to correctly identify all discrepancies. This protocol mainly consists of following steps. First, both sender and receiver run DIP with a guess of discrepancies size, $d'$, obtaining a rational function $g'(t)$. Then, a probabilistic equality test is employed to check if $g'(t)$ and $\mathcal{P}_P(t)/\mathcal{P}_S(t)$ agree on $\tau$ random evaluation points. If $g'(t)$ passed this probabilistic equality test, PIP-OC terminates. Otherwise, receiver sends a failure notification to sender to trigger another round with increased $d'$.

According to the theorem of monic rational functions in [32], the probability that PIP-OC terminates with $g'(t) \neq \mathcal{P}_P(t)/\mathcal{P}_S(t)$ is no more than $d\rho^\tau$. Thus if we need PIP-OC failure probability to be less than $\delta$, the $\tau$ should be:

$$\tau \geq \lceil \log_\rho(\frac{\delta}{d}) \rceil \geq \lceil \log_\rho(\frac{\delta}{|P|+|S|}) \rceil \tag{3}$$

where $\rho = \frac{|P|+|S|-1}{2^{b_t}}$. For example, given $P$ and $S$ are 64-bit strings lists of size 200,000,000, $\tau = 1$ is enough to ensure $\delta$ is at most 0.01.

The pseudocode is shown in Algorithm 1. Since the communication complexity is the primary concern here, the evaluation values are sent one by one. In order to reduce transmission of random evaluation points, a pseudo-random number generator can used to generate those points on both sides.

*Analysis:* Since pesudo-random number generator is employed, we do not need to transfer those random points information. Also as indicated in equation 3, $\tau = 1$ applies for most of practical cases. But we need to transfer $\tau$ extra check values and $(d + \tau)$ bits feedbacks to sender. Thus the communication complexity of PIP-OC is

$$((b_t+1)d+b)+(b_t+1)\tau+(d+\tau)=(b_t+2)(d+1)+b_t.$$

Meanwhile, since DIP may be executed at most $d$ times in Algorithm 1, the decoding complexity of PIP-OC is $\mathcal{O}(d \cdot d^3) = \mathcal{O}(d^4)$.

**Algorithm 1** PIP-OC

1: Initialize receiver's values list $\mathcal{L} \leftarrow \emptyset$.
2: Sender transmits $\mathcal{P}_S(t_1)$ at random point $t_1$ as evaluation point.
3: Receiver adds $\mathcal{P}_S(t_1)$ to list $\mathcal{L}$.
4: Sender transmits $\mathcal{P}_S(t')$ at random point $t'$ and marks it as check point.
5: Receiver run DIP with list $\mathcal{L}$.
6: **if** $g'(t) = \mathcal{P}_P(t)/\mathcal{P}_S(t)$ at checkpoint **then**
7: Return zeros of numerator and denominator of $g'(t)$.
8: **else**
9: Receiver adds $\mathcal{P}_S(t')$ to list $\mathcal{L}$.
10: Goto Line 4.
11: **end if**

## VI. PROBABILISTIC IDENTIFICATION PROTOCOL WITH OPTIMIZED ROUNDS

While PIP-OC is designed for achieving minimal communication complexity, it has the disadvantage of requiring $(d+\tau)$ communication rounds. Thus, in this section we focus on how to identify discrepant tags with optimized rounds and thereby propose Probabilistic Identification Protocol with Optimized Rounds (PIP-OR). In some cases, the number of communication rounds is an important metric in communication system, since undesirable network delay might be caused by long distance communication (e.g., in EPCglobal Network), leading to long communication time. The key difference between PIP-OR and PIP-OC is that in PIP-OR we reduce the number of rounds to $\lceil \log_\lambda(d+\tau) \rceil$ through incrementally increasing the number of evaluation points by a factor of $\lambda$. The pseudocode is shown in Algorithm 2.

**Algorithm 2** PIP-OR

1: $d' \leftarrow 1$.
2: Sender transmits $\mathcal{P}_S(t_1), ..., \mathcal{P}_S(t_{d'})$ at random point $t_1, ..., t_{d'}$ as evaluation points and $\mathcal{P}_S(t')$ at random point $t'$ as check point.
3: $\mathcal{L} \leftarrow \emptyset$.
4: Receiver adds all evaluation points to list $\mathcal{L}$.
5: Receiver run DIP with list $\mathcal{L}$.
6: **if** $g'(t) = \mathcal{P}_P(t)/\mathcal{P}_S(t)$ at checkpoint **then**
7: Return zeros of numerator and denominator of $g'(t)$.
8: **else**
9: $d' \leftarrow d' \cdot \lambda$.
10: Goto Line 2.
11: **end if**

*Analysis:* According to Algorithm 2, we can compute the communication complexity of PIP-OR as:

$$\sum_{i=1}^{\alpha} \beta_i + b_t + \alpha \tag{4}$$

, where $\beta_i = (b_t + 1)(\lambda^{i-1} + 1)$, $\alpha = \lceil \log_\lambda(d+1) \rceil$. $\beta_i = (b_t+1)(\lambda^{i-1}+1)$ stands for the total bits of transferred sample points in $i$-th round, including evaluation points and check point. $b_t$ bits is counted for one-time transferring the size of ship list. $\lceil \log_\lambda(d+1) \rceil$ is the total acknowledgement bits to indicate whether passing equality test. It it easy to see that $\sum_{i=1}^{\alpha} \beta_i$ is the dominating part in communication complexity. Therefore, we seek to minimize it as follows.

We use $\bar{F}$ to denote the communication complexity of DIP with a priori $d$, excluding $b$ bits for transferring the size of ship list. Also $F(n)$ is used to denote the sum of $\beta_1, \ldots \beta_n$,

$$F(n) = \beta_1 + \cdots + \beta_n = (b_t + 1)(\frac{\lambda^n - 1}{\lambda - 1} + n).$$

It is easy to see that $\bar{F} \geq \beta_{n-1}$ since $n$ is the first round that passes equality test. Thus, we can have

$$\frac{F(n)}{\bar{F}} \leq \frac{\frac{\lambda^n - 1}{\lambda - 1} + n}{\lambda^{n-2} + 1}. \tag{5}$$

Since the right side of equation 5 is maximized as $n$ approaches infinity, it leads to $\frac{F(n)}{\bar{F}} \leq \frac{\lambda^2}{\lambda-1} = \gamma(\lambda)$. By computing the derivative of $\gamma(\lambda)$, we find it achieves minimum value at $\lambda = 2$. That is to say, if we choose $\lambda = 2$ which leads to $F(n) \leq 4\bar{F}$, we can know that the communication complexity of PIP-OR is approximately no more than 4 times of DIP which requires $d$ as a priori.

## VII. PROBABILISTIC IDENTIFICATION PROTOCOL WITH ESTIMATION

PIP-OR works well to reduce the number of communication rounds to $\lceil \log_\lambda(d + \tau) \rceil$. But its value is still not so impressive especially when discrepancies size $d$ is large, such as $2^{32}$. And someone may wonder if there is possible to finish discrepancy identification with just one round? To answer this question, we design a Probabilistic Identification Protocol with Estimation (PIP-E). The central idea of PIP-E is a hierarchical estimator that can efficiently approximate discrepant tag population.

Our estimator is built on top of FM-Sketch [34]. In traditional FM-Sketch, each bit of the data structure is either 0 or 1. The $j$-th bit is 1 meaning that there is at least 1 element is sampled with $2^{-j}$ probability. For example, assume we have 8 distinct elements in the list, the 3-rd bit is set to 1 with sampling probability at $1/8$. Hence, the size of the list is estimated as $2^3 = 8$. We however find that FM-Sketch can only estimate the size of the all list, but not the population of discrepant tags. This is because no information regarding to the common elements is include in the original FM-Sketch data structure. Hence, we design a novel hierarchical estimator to do this job.

We describe the how the hierarchical estimator works. In the beginning, we group tags geometrically into $b_t$ layers.

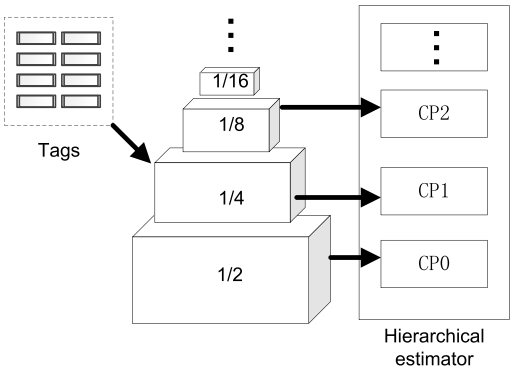


**FIGURE 4.** Hierarchical estimator construction.

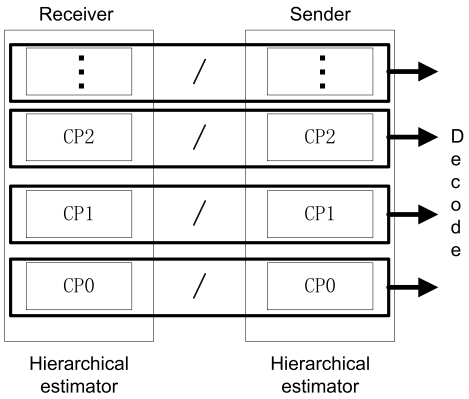


**FIGURE 5.** Hierarchical estimator decoding.

For the $j$-th layer, it roughly corresponds to $2^{-(j+1)}$ elements. We follow the convention that using the number of leading zeros based on the binary form to perform such grouping. We then compute the characteristic polynomial (CP$i$) for each layer with $\theta$ sample points. An example is given in Figure 4. Let's see a toy example. Suppose $b_t = 4$, element $5 = (00101)_2$ would be grouped in CP2 and element $8 = (01000)_2$ would be in CP1.

As demonstrated in Figure 5, the building block of decoding for the hierarchical estimator is running DIP for each level. Then probabilistic equality tests are used to do comparison, which is the same as PIP-OC and PIP-OR. Hence, we can obtain accurate discrepancies information at each level with the help of accurate estimation results. We show pseudocodes in Algorithm 3.

Now we show that by using the above hierarchical estimator, we can realize discrepancy identification using just a single round communication. First, the receiver computes its hierarchical estimator and transmits it to the sender. The sender, upon receiving the estimator, can approximate the discrepant tag population, $d$, using its own estimator. Later characteristic polynomials evaluated over $d$ points are sent back to the receiver. Last, the receiver use interpolation to extract all the discrepant tag information. There is one thing worth noting. The geometric partition mentioned earlier needs a uniformly random hash. In our system, we use well-known $t$-wise independent hash functions for implementation.

**TABLE 2.** Schemes comparison for discrepant tag identification.

| Scheme | Prior | Communication complexity | Round complexity |
|---|---|---|---|
| WP | No | $b_t\|S\|$ | 1 |
| BFS | No | $b_t * \|\Delta S\| + \mathcal{O}(\|P\| + \|S\|)$ | 1 |
| DIP | Yes | $(b_t + 1)(d + 1) - 1$ | 1 |
| PIP-OC | No | $(b_t + 2)(d + 1) + b_t$ | $d + 1$ |
| PIP-OR | No | $(b_t + 1)(\lambda(d + 1) + \lceil \log_\lambda(d + 1)\rceil) + b_t + \lceil \log_\lambda(d + 1)\rceil$ | $\lceil \log_\lambda(d + 1)\rceil$ |
| PIP-E | No | $(b_t + 1)((\theta + 1)(b_t + 1) + d + 1) - 1$ | 1 |

**Algorithm 3** Hierarchical Estimator Decoding in PIP-E

1: Encode ship list and physical list into CP$i_1$ and CP$i_2$, $0 \leq i \leq b_t - 1$.
2: Initialize the number of different elements $c \leftarrow 0$.
3: **for** $i = b_t - 1$ to 0 **do**
4: **if** $i >= 0$ and CP$i_1$/CP$i_2$ passes probabilistic equality test **then**
5: $c \leftarrow c+$ number of zeros in numerator and denominator of CP$i_1$/CP$i_2$.
6: **else**
7: Return $d \leftarrow 2^{i+1} \cdot c$.
8: **end if**
9: **end for**

*Analysis:* The communication overhead for each layer of the estimator is $\theta$ points plus a check point, which is

$$\theta(b_t + 1) + b_t + 1.$$

Hence, the total overhead of PIP-E is the overhead of the estimator for $b_t$ levels and the overhead of DIP,

$$(b_t + 1)(\theta b_t + b_t + d + 1) - 1. \tag{6}$$

For the rigorous proof, please refer to [1]. In our real experiments, we use 96 levels and 40 points for each level. Such a estimator can take up to $2^{96}$ tags and discover the discrepant tags with high probability. This way, PIP-E Thus, PIP-E can achieve constant communication overhead for discrepancy identification.

## VIII. EVALUATION

In this part, first we present detailed theoretical analysis and comparison to show advantages of our proposed protocols. Then we show the results of extensive experiments that examine the efficiency of our schemes. We mainly focus on communication overhead: the amount of traffic and time.

### A. ANALYTICAL COMPARISON

#### 1) BLOOM FILTER BASED APPROACH

In fact, the bloom filter, a well-known data structure for compressed membership inquiry, can also be used to solve discrepant tag identification. We briefly describe the bloom filter based solution (BFS) as follows:

1) The receiver computes its bloom filter, $BF_1$, and sends it to the sender.
2) The sender discover the elements that are not included in $BF_1$ and put them into the set $\Delta S$, which is transmitted back to the receiver along with its own bloom filter, $BF_2$.
3) The receiver finds out the elements that are not included in $BF_2$ and put them into the set $\Delta P$.
4) $\Delta P$ and $\Delta S$ are the results that contain discrepant tags.

According to [35], the optimal bloom filter length should follow:

$$l \geq -\frac{h}{\ln(1 - p^{-1/h})}|W|, \tag{7}$$

where $h$ is the number of hash functions, $p$ is the false positive probability and $|W|$ is the size of original data list. Thus, the communication complexity of (BFS) is

$$b_t * |\Delta S| + \mathcal{O}(|P| + |S|). \tag{8}$$

#### 2) SCHEMES COMPARISON

Table 2 compares following protocols: traditional wholesale protocol, bloom filter based protocol, deterministic identification protocol, and our other three probabilistic protocols. The comparison is based on three important metrics: (i) whether requiring a priori; (ii) communication complexity which is the total bits transferred in network; (iii) round complexity which accounts for how many round trip communication.

- **WP:** Although its communication rounds is optimal, the complexity of WP grows linearly with the size of ship list which does not scales well with growing larger volume of RFID data.
- **BFS:** It consumes less data bits than WP, but it still suffers from scalability issue since its communication complexity grows linearly with the size of ship list and physical list even with optimal parameters for bloom filter.
- **DIP:** It scales well with discrepant tag population, but requiring a priori $d$. Thus we use its complexity as the *lower bound* of discrepant tag identification without knowing discrepancies size.
- **PIP-OC:** Its communication complexity is very close to the lower bound. The communication rounds, however, requires $d + 1$. Thus it may not be suitable for high latency network.
- **PIP-OR:** Its communication rounds is log reduced compared to PIP-OC. Its communication complexity is also

**TABLE 3. Main parameters in experiments.**

| Parameter | Value |
|---|---|
| Number of warehouses | 5 |
| Pallet injection frequency | 10 per second |
| Cases per pallet | 10 |
| Items per case | 100 |
| Discrepancies ratio($\eta$) | [0.01, 0.99], default 0.01 |
| Ship list size ($N$) | [10,000, 100,000], default 50,000 |
| Round trip time (RTT) | [1, 1000]ms, default 10ms |
| Bandwidth ($B$) | default 20Mbit/s |
| $p$ in bloom filter | default 0.01% |

bounded as roughly 4 times of DIP as shown in section V. A nice balance between communication complexity and round complexity is achieved.

- **PIP-E:** With the help of a novel hierarchical estimator, its round complexity is optimal at the expense of some additional data transmission.

Now we summarize the relationships among DIP, PIP-OC, PIP-OR, and PIP-E. DIP is most suitable for the cases where the size of discrepant tags, $d$, are known. However, for most of the time, such sizes are unknown in systems and could even vary significantly. Then in scenarios where $d$ is uncertain, we can resort to PIP-OC, PIP-OR, and PIP-E. Specifically, PIP-OC is fit for the low bandwidth and low latency network. But as a number of rounds, $d + 1$, are needed for PIP-OC, its latency could become an issue. So for networks where latency is a concern, we can use PIP-OR or PIP-E depending on the latency requirements. PIP-OR is more suitable for intermediate latency networks while PIP-E is the best for high latency networks because PIP-E only has one round. Note that PIP-OC has the lowest transmission data, i.e., no single scheme can outperform others in terms of both data and time. Therefore, the network administrator can make various tradeoffs between transmission data (data complexity) and transmission time (round complexity) by choosing from PIP-OC, PIP-OR, and PIP-E. Next, we will show how these protocols behave in various network conditions.

### B. EXPERIMENTAL COMPARISON

In order to show the effectiveness of our schemes, we use a customized simulator based on CSIM [36]. Our detailed experimental setup and implementation are as follows.

#### 1) EXPERIMENTAL SETUP

Table 3 includes the main parameters in our tests. We have set the number of warehouses as 5. For each source warehouse, we manually inject pallets of cases and make it move across warehouses. We also randomly remove, inject, and replace the tag sets at each checkpoint but keep the ratio at $\eta$. After each scanning, the receiver will perform discrepant tag identification based on the EPCglobal network for all kinds of inquiries. Currently, due to the lack of well-known RFID data for test, we choose to generate random 100,000,000 tags and store them in the backend database. For finite field arithmetic operations, we employ Victor Shoupąŕs Number Theory Library [37]. We do each test for 100 times and report averages. Note that the value of the bandwidth can vary in different systems and we use 20 Mbit/s as a default setting because it comes from a practical EPCglobal system of a logistics corporation.

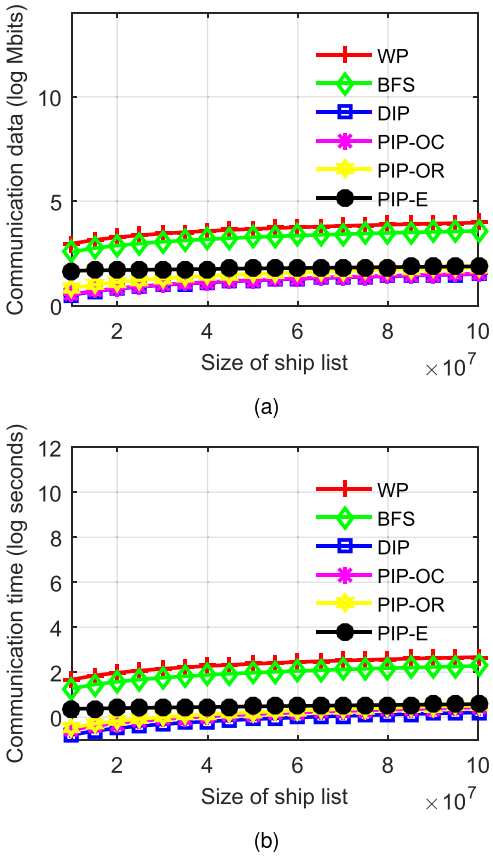


**FIGURE 6. Communication overhead with RTT=1ms : (a) Communication data VS varying size of ship list; (b) Communication time VS varying size of ship list.**

#### 2) VARYING SIZE OF SHIP LIST

We set the range of ship data size between 10,000 and 100,000 and set RTT at 1 ms. For all the other parameters, we use default values. We have learning several observations from such a very low latency network as shown in Figure 6. First of all, all the three probabilistic protocols have lower traffic and communication time than WP and BFS do. For instance, the communication data of PIP-OC is merely 0.36% and 0.91% of WP and BFS respectively, when size of ship list is 10,000. The primary reason is that the communication complexity of WP and BFS grow linearly with overall tag population. In contrast, the probabilistic protocols grows linearly with the discrepancy tag population. This confirms our analytical results in Table 2. Second, as shown in Figure 6a, PIP-OC is the closest to lower bound (DIP) for data size,

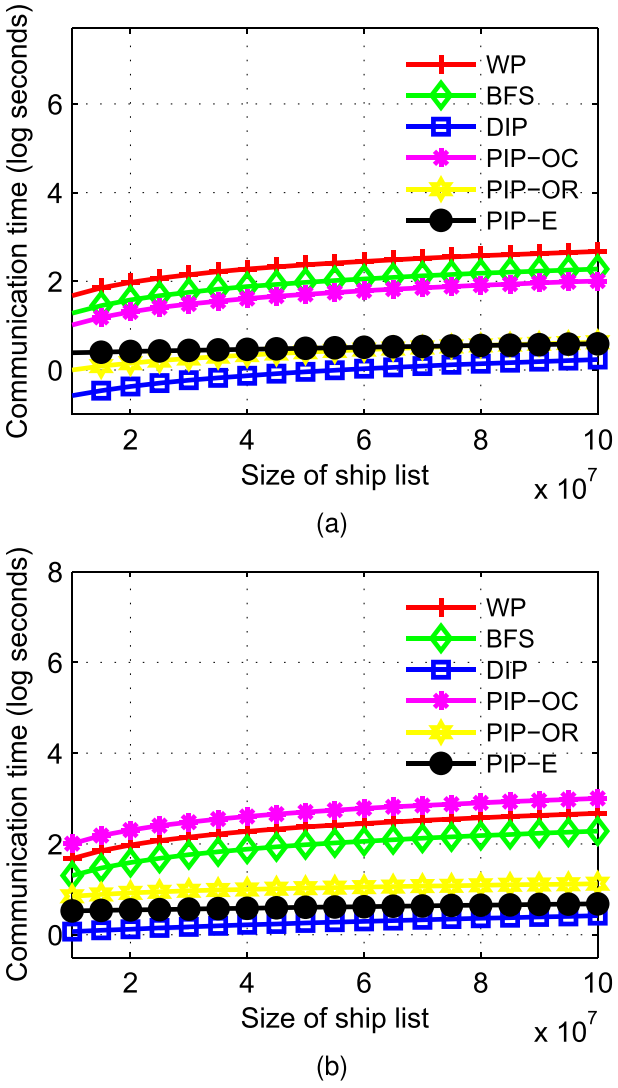


**FIGURE 7.** Communication time under different latencies: (a) intermediate latency network, RTT=100ms; (b) high latency network, RTT=1000ms.

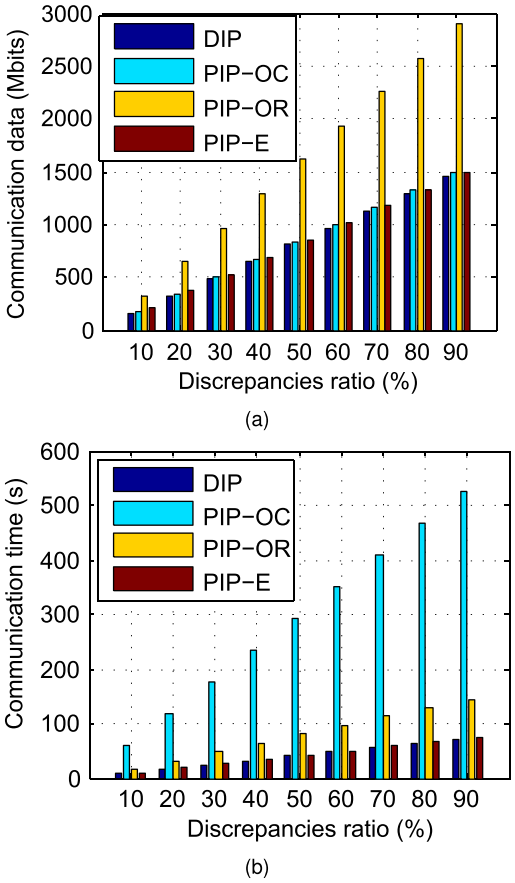


**FIGURE 8.** Communication overhead under different discrepancies ratio: (a) Communication data VS discrepancies ratio; (b) Communication time VS discrepancies ratio.

since it only incurs a little more bits for acknowledgement. The communication data volume of PIP-OR and PIP-E are higher than PIP-OC since they trade some more bits for lesser round complexity. But in cases where the network latency is low, the advantage of having less rounds becomes less noticeable in Figure 6b. Third, although communication cost of all protocols are getting higher as the size of discrepant is increasing, PIP-E is the most unsensitive of them. This is because that PIP-E has a well-designed difference estimator.

### 3) VARYING NETWORK LATENCY

In addition, we examine the performance of our protocols with different latencies, namely RTT=100ms, which stands for intermediate latency networks and RTT=1000ms which represents high latency networks. The results are shown in Figure 7. Not surprisingly, PIP-OC's performance is not good in intermediate latency network as illustrated in Figure 7a and is getting worse in high latency network since its communication time is even more than WP and BFS in Figure 7b. PIP-OR is the best protocol in intermediate latency network as it is well-balanced between communication complexity and round complexity. In scenarios where networks have high latency, PIP-E is the best among all the protocols. In particular, the communication time of PIP-E is only 36.2% and 0.47% of PIP-OR and PIP-OC, respectively, when RTT=1000ms and $N = 100,000$. Combining results in Figure 6b and Figure 7, we can conclude that our three probabilistic protocols achieve various trade-offs in communication complexity and round complexity. Thus, if properly chosen, our probabilistic protocols are able to adapt to various network conditions.

### 4) VARYING DISCREPANCIES RATIO

In this part, we evaluate the impact of discrepancies ratio to our protocols. The discrepancies ratio is set from 10% to 90% and other parameters are default. The results are shown in Figure 8. For communication data, all protocols' communication complexities grows as discrepancies ratio is increasing. The performances of PIP-OC and PIP-E are close to

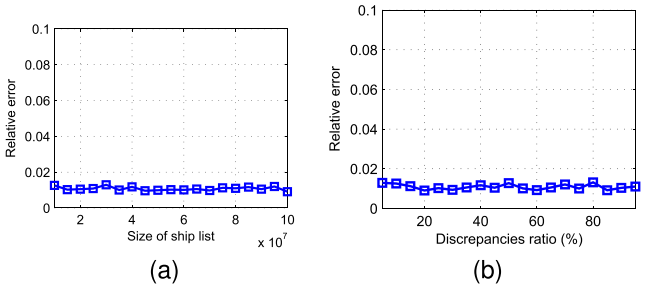


**FIGURE 9.** Relative standard error under different ship size and discrepancies ratio: (a) Relative standard error VS different ship size; (b) Relative standard error VS discrepancies ratio.

lower bound (DIP) while PIP-OR is always the highest. This is mainly due to larger number of rounds of PIP-OR ($\lceil \log_{\lambda}(d+1) \rceil$) caused by increasing discrepant tag size, whereas PIP-E uses constant size of estimator and fixed one round-trip whatever the size of discrepant tags. For communication time, PIP-OC is the worst due to RTT=10ms. At this time, PIP-OC are unable to catch up with DIP as discrepancies ratio increases, while PIP-E still keeps close to the lower bound. Therefore, we can see that even if low latency network (RTT=10ms), PIP-E should be the first choice when discrepancies ratio is relative high.

#### 5) ESTIMATOR OF PIP-E

At last, we investigate PIP-E's estimation accuracy. We employ relative error ($\varepsilon$), defined as

$$\varepsilon = \frac{\hat{n} - n}{n},$$

where $\hat{n}$ is the estimation result and $n$ is the ground truth number. The tests are done with varying sizes of the ship list between 10,000 and 100,000 as in Figure 9a. The discrepancy ratio is set between 5% and 95% as shown in Figure 9b. Our main observation is that $\varepsilon$ is always around 0.01 for most of the tests, which helps explain why our PIP-E is highly efficient.

## IX. CONCLUDING REMARKS

In this paper, we have proposed several efficient protocols for efficiently identifying discrepant tags. The key idea of our protocols is to make the communication overhead scale to the discrepant tag population, but not the overall tag population. In particular, we propose deterministic protocol to fast identify discrepancies using polynomials, requiring the size of discrepant tags as a priori. To remove this constraint, we design three probabilistic protocols each of which is built on top of another to progressively achieve balance between communication complexity and round complexity. Through meticulous analysis and extensive simulations, we demonstrate our proposals significantly outperform former wholesale protocol and bloom filter based solution. In addition, our protocols are able to adapt to various network conditions and system parameters, such as latency, discrepant ratio and ship list size.

## REFERENCES

[1] C. Gu, W. Gong, and A. Nayak, "Identifying discrepant tags in RFID-enabled supply chains," in *Proc. WASA*, 2016, pp. 162–173.
[2] D. Benedetti, G. Maselli, and C. Petrioli, "Fast identification of mobile RFID tags," in *Proc. IEEE MASS*, Oct. 2012, pp. 65–74.
[3] R. Kumar, T. F. La Porta, G. Maselli, and C. Petrioli, "Interference cancellation-based RFID tags identification," in *Proc. ACM MSWiM*, 2011, pp. 111–118.
[4] "Wal-Mart's RFID refresh," *RFID J.*, 2007. [Online]. Available: http://www.rfidjournal.com/articles/view?6842
[5] D. Lunn, " Employee theft: Eliminate the opportunity guest series part 1," Tech. Rep.
[6] J. Liu, B. Xiao, K. Bu, and L. Chen, "Efficient distributed query processing in large RFID-enabled supply chains," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 163–171.
[7] H. Ziekow, B. Fabian, and C. Müller, "High-speed access to RFID data: Meeting real-time requirements in distributed value chains," in *Proc. Workshops. Move Meaningful Internet Syst. (OTM)* , 2009, pp. 142–151.
[8] J. Yu, L. Chen, R. Zhang, and K. Wang, "On missing tag detection in multiple-group multiple-region RFID systems," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1371–1381, May 2017.
[9] 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.
[10] W. Gong, J. Liu, and Z. Yang, "Efficient unknown tag detection in large-scale RFID systems with unreliable channels," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2528–2539, Aug. 2017.
[11] F. Zhu, B. Xiao, J. Liu, and L.-J. Chen, "Efficient physical-layer unknown tag identification in large-scale RFID systems," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 283–295, Jan. 2017.
[12] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MobiCom*, 2006, pp. 322–333.
[13] Y. Zheng, M. Li, and C. Qian, "PET: Probabilistic estimating tree for large-scale RFID estimation," in *Proc. IEEE ICDCS*, Jun. 2011, pp. 37–46.
[14] M. Shahzad and A. X. Liu, "Every bit counts: Fast and scalable RFID estimation," in *Proc. ACM MobiCom*, 2012, pp. 365–376.
[15] W. Gong, J. Liu, K. Liu, and Y. Liu, "Toward more rigorous and practical cardinality estimation for large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1347–1358, Jun. 2017.
[16] T. Li, S. S. Wu, S. Chen, and M. C. K. Yang, "Generalized energy-efficient algorithms for the RFID estimation problem," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1978–1990, Dec. 2012.
[17] Q. Xiao, B. Xiao, S. Chen, and J. Chen, "Collision-aware churn estimation in large-scale dynamic RFID systems," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 392–405, Feb. 2017.
[18] Z. Zhou, B. Chen, and H. Yu, "Understanding RFID counting protocols," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 312–327, Feb. 2016.
[19] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. ACM MOBIHOC*, 2010, pp. 1–10.
[20] C. C. Tan, B. Sheng, and Q. Li, "How to monitor for missing RFID tags," in *Proc. IEEE ICDCS*, Jun. 2008, pp. 295–302.
[21] W. Luo, S. Chen, T. Li, and Y. Qiao, "Probabilistic missing-tag detection and energy-time tradeoff in large-scale RFID systems," in *Proc. ACM MobiHoc*, 2012, pp. 95–104.
[22] L. Bolotnyy and G. Robins, "Physically unclonable function-based security and privacy in RFID systems," in *Proc. IEEE PERCOM*, Mar. 2007, pp. 211–220.
[23] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Security in Pervasive Computing*. Los Alamitos, CA, USA: IEEE Comput. Soc., 2004, pp. 50–59.
[24] L. Lu, J. Han, L. Hu, Y. Liu, and L. M. Ni, "Dynamic key-updating: Privacy-preserving authentication for RFID systems," in *Proc. IEEE PERCOM*, Mar. 2007, pp. 13–22.
[25] L. Lu, J. Han, R. Xiao, and Y. Liu, "ACTION: Breaking the privacy barrier for RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1953–1961.
[26] L. Lu, Y. Liu, and X. Li, "Refresh: Weak privacy model for RFID systems," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[27] W. Gong, I. Stojmenovic, A. Nayak, K. Liu, and H. Liu, "Fast and scalable counterfeits estimation for large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1052–1064, Apr. 2016.
[28] C.-H. Lee and C.-W. Chung, "Efficient storage scheme and query processing for supply chain management using RFID," in *Proc. ACM SIGMOD*, 2008, pp. 291–302.

[29] Z. Cao, C. Sutton, Y. Diao, and P. Shenoy, "Distributed inference and query processing for RFID tracking and monitoring," *Proc. VLDB Endowment*, vol. 4, no. 5, pp. 326–337, 2011.
[30] M. Blum and S. Kannan, "Designing programs that check their work," *J. ACM*, vol. 42, no. 1, pp. 269–291, 1995.
[31] Y. Minsky, A. Trachtenberg, and R. Zippel, "Set reconciliation with nearly optimal communication complexity," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2213–2218, Sep. 2003.
[32] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, vol. 12. New York, NY, USA: Springer-Verlag, 2002.
[33] R. E. Zippel, *Effective Polynomial Computation*. Boston, MA, USA: Springer, 1993.
[34] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," *J. Comput. Syst. Sci.*, vol. 31, no. 2, pp. 182–209, 1985.
[35] A. Z. Broder, "On the resemblance and containment of documents," in *Proc. Compress. Complex. Sequences*, Jun. 1997, pp. 21–29.
[36] *CSIM*. Accessed: Jan. 5, 2018. [Online]. Available: http://www.csim.com/
[37] *NTL*. Accessed: Jan. 5, 2018. [Online]. Available: http://www.shoup.net/ntl/

**CAIDONG GU** received the master's degrees in computer science technology from the University of Liaoning, China. He is currently a Professor with Suzhou Vocational University. His current research interests include radio frequency identification and data mining.