Fast Counting the Key Tags in Anonymous RFID Systems

Xiulong Liu*, Keqiu Li*§, Heng Qi*, Bin Xiao† and Xin Xie*
*School of Computer Science and Technology, Dalian University of Technology, China
†Department of Computing, The Hong Kong Polytechnic University, Hong Kong
§Corresponding Author is Keqiu Li
Email: {xiulongliudut, likeqiu, qhclement, xiexin0211}@gmail.com, csbxiao@comp.polyu.edu.hk

***Abstract*—In RFID-enabled applications, we may pay more attention to key tags instead of all tags. This paper studies the problem of *key tag counting*, which aims at estimating how many key tags in a given set exist in the current RFID system. Previous work is slow to solve this new problem because of the serious interference replies from the large number of ordinary (i.e., non-key) tags. However, time-efficiency is an important metric for the fast tag cardinality estimation in a large-scale RFID system. In this paper, we propose a singleton slot-based estimator, which is time-efficient because the RFID reader only needs to observe the status change of expected singleton slots of key tags instead of the whole time frame. In practice, the ratio of key tags to all current tags is small for "key" members should be rare. As a result, even when the whole time frame is long, the expected singleton slot number is limited and the running of our protocol is fast to achieve estimation accuracy. Rigorous theoretical analysis shows that the proposed protocol can provide guaranteed estimation accuracy to end users. We conduct simulations and implement a prototype of our protocol to verify its efficiency and deployability.**

I. Introduction

Radio Frequency Identification (RFID) is a wireless technology that uses a RFID reader to monitor or identify the objects or even humans by reading the attached tags. It has promising prospects in various applications such as supply chain management [1], access control [2], localization [3], object tracking [4]. In a large-scale RFID system containing thousands of tags, the manager may only care about the **key** tags (e.g., the tags attached to expensive jewelries or encapsulated in cards of key visitors) instead of all tags. A primary question is how many key tags there are in the current system that also contains a large number of ordinary (non-key) tags. The counting result, which can provide the information about the popularity of key items or the attendance of key visitors, is of practical importance. Hence, this paper studies the new problem of **key tag counting**—estimating the cardinality of key tags that are present in the system.

The key tag counting problem is formulated as follows. We use $S_K = \{x_1, x_2, ..., x_k\}$ to represent the set (list) of k key tags that we are interested in. S_K is known in advance. We use $S_C = \{y_1, y_2, ..., y_c\}$ to denote the current (actual) set of tags in the system. In reality, S_C is not known in prior either for the privacy reason, or because it is not easy to get, particularly in dynamic RFID systems (the tagged objects or humans frequently move in or out). In such an anonymous RFID system, there are three types of tags as illustrated in Fig. 1:

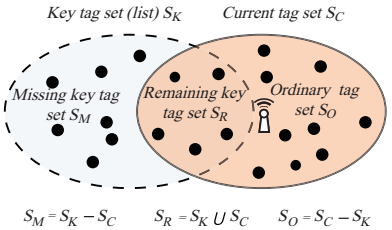

Fig. 1: Exemplifying the problem of key tag counting.

- *Missing key tag set* S_M: the tags that we are interested in but are not present in the current system. $S_M = S_K - S_C$ and $|S_M|$ is denoted as m.
- *Remaining key tag set* S_R: the present tags that we are interested in. $S_R = S_K \bigcap S_C$ and $|S_R|$ is denoted as r.
- *Ordinary tag set* S_O: the present tags that we are not interested in. $S_O = S_C - S_K$ and $|S_O|$ is denoted as o.

The key tag counting problem is how to produce an estimate $\hat{r}$ for r, so that $P\{|\hat{r} - r| \leq \alpha \cdot r\} \geq \beta$, where $\alpha \in (0, 1)$ is the allowed relative error and $\beta \in (0, 1)$ is the required confidence level. Here, α and β indicate the estimation accuracy, and should be specified by the end users in prior. For simplicity, we refer to $\hat{r}$ as an (α, β) estimate of r.

Prior Art and Limitation. In the following, we briefly review the related work and point out their limitations when facing the problem of key tag counting.

1) Pure Identification Protocols: An immediate method is to identify all tags in S_C, and then we can get the exact r by comparing S_K and S_C. There have been already a number of schemes proposed for solving the problem of tag identification [5] [6] [7] [8] [9] [10]. These schemes, however, suffer two fatal drawbacks: (i) The processing time rapidly grows as the number of RFID tags increases. According to the RFID standard ISO-18000, the average identification throughput is just about 100 tags per second [11]; (ii) Privacy leaking due to the transmission of tag IDs (as plaintext) in the air.

2) Tag Search Protocols: The literature [11] [12] exploited the Bloom filter technique to search the exact tags in S_R. Although the methods in [11] [12] have improved a lot than

the pure identification protocols, they are still of low time efficiency to solve the key tag counting problem. The intuitive reason is that searching the exact tags consumes more time than just counting (or estimating) the cardinality of tags.

3) Cardinality Estimation Protocols: Most of the existing estimation protocols [13] [14] [15] [16] [17] [18] [19] [20] [21] can only estimate the cardinality of S_C, and thus are not able to count the key tags. Schemes in [22] [23] are not efficient to solve the problem of key tag counting because they require the reader to observe the whole time frame.

This paper thoroughly studies the problem of key tag counting and the main contributions are summarized as follows.

(1) **Proposing Novel Singleton-based Estimators.** We first propose a *Basic Key tag Counting (B-KC)* protocol, whose unique feature different from prior work is that *the reader only needs to observe the expected singleton slots corresponding to key tags instead of the whole time frame*. The expected empty/collision slots are not used, and are directly skipped for saving time. To be more scalable, we exploit the sampling idea on B-KC to propose the *Sampling-based Key tag Counting (S-KC)* protocol, in which only sample tags participate in the estimation process. In fact, B-KC is a special case of S-KC when the sampling probability is set to 1.

(2) **Mathematically Investigating the Accuracy of the Proposed Estimators.** We leverage mathematical tools such as Taylor Series Expansion [24] and Central Limit Theorem [25] to theoretically give the answer to an important question—*how many frames are adequate to produce an* (α, β) *estimation result*.

(3) **Proposing the Early Termination Tactic.** We observe that the minimum frame number $\aleph$ is actually *over calculated*. The underlying reason is that the expression of $\aleph$ contains two variables u and d which are unknown in prior. To guarantee the predefined accuracy for any actual values of u and d, we have to use their extreme values (i.e., u_{max} and d_{max}) to configure the frame number, which leads to the over calculation of $\aleph$. As a result, the performance of S-KC is far from its ideal case (i.e., assuming u and d are known in prior). To fill this performance gap, we leverage the *three-sigma rule* [26] to give the *tighter* bounds of u and d after each frame. Then, backend server is able to dynamically determine whether to terminate the estimation process. We refer to this technique as *early termination*, which can make the performance of S-KC closer to its ideal case.

(4) **Simulation and Prototype.** We conduct simulations to evaluate the performance of S-KC in a large-scale RFID system that contains thousands of tags. Simulation results show that S-KC runs several times faster than the recent work [11] [12] [22] [23]. Moreover, we use nRF24LE1, the highly integrated ultra low power 2.4GHz RF System-on-Chip (SoC), to implement a prototype of our S-KC, which reveals the deployability of our protocol.

The rest of this paper is organized as follows. Section II presents the system model. We propose the B-KC and S-KC and present the theoretical analyzes in Sections III and IV, respectively. In Section V, extensive simulation experiments are conducted to evaluate the performance of the proposed protocol. The related work is reviewed in Section VI. Finally, this paper is concluded in Section VII.

II. System Model

In this paper, we consider a RFID system that consists of three components: a backend server, a single (or multiple) reader(s), and a large number of tags. The backend server is able to store a large amount of data and perform complex computations. The reader has a dedicated power source, and is connected to the backend server via high data rate communication link. For the purpose of clarity, we first study the single-reader scenario, and assume the reader has adequate interrogating ranges to probe all tags [15] [17] [27]. Then, we generalize the proposed protocol to the multi-reader case. In the rest of this article, the reader and the backend server are regarded as an integrated unit, which is still referred to as reader for simplicity. Each tag has a unique 96-bit ID according to EPC C1G2 standard [28], and is used to identify an individual object. The communication between the reader and tags are in the Reader Talks First (RTF) mode [17], i.e., the reader queries the tags first, and the tags respond over a shared wireless medium.

The reader continuously sends synchronization signals to create a slotted time frame. And the tags contend for slots to transmit responses. We classify the time slots into three categories: *empty slot* indicates there is no tag response in this slot; *singleton slot* denotes that only one tag responds in this slot; and *collision slot* means two or more tags simultaneously transmit in this slot and collision happens. To distinguish an empty slot from a non-empty slot, 1-bit response is adequate. On the other hand, at least 10-bit response is required to verify a collision slot [29]. Hence, based on their length and use, slots can also be classified into: *short-response slot* that is used to transmit 1-bit information; *long-response slot* supporting transmission of 10-bit information; and *tag slot* that can be used to transmit 96-bit tag ID [29].

According to the specification of the Philips I-Code system [30], the wireless transmission rate from a tag to a reader is 53 Kb/s, that is, it takes a tag $18us$ to transmit 1-bit data. And the rate from a reader to a tag is 26.5 Kb/s, that is, transmission of 1-bit data to tags requires $37.7us$. For simplicity, we assume the transmission rate from the reader to tags and that from tags to the reader are the same. We choose the relatively low rate 26.5 Kb/s as the common transmission rate, and then the following slot settings can support transmission from both reader to tags and vice versa. Any two consecutive transmissions (from a tag to a reader or vice versa) are separated by a waiting time $\tau_0 = 302us$ [19] [29]. And thus, the time of a short-response slot, t_{short}, is set to $37.7us + 302us \approx 0.4ms$; the time of a long-response slot, t_{long}, is set to $37.7us \times 10 + 302us \approx 0.7ms$; and the time of a tag slot, t_{tag}, is set to $37.7us \times 96 + 302us \approx 4ms$. Table I lists the symbols used in this paper.

TABLE I: Symbols used in the paper

Symbols	Descriptions
S_M	missing tag set.
m	cardinality of S_M, $m = \|S_M\|$.
S_R	remaining tag set.
r	cardinality of S_R, $r = \|S_R\|$.
S_O	ordinary tag set.
o	cardinality of S_O, $o = \|S_O\|$.
S_K	key tag set, $S_K = S_M \bigcup S_R$.
k	cardinality of S_K, $k = \|S_K\|$, $k = m + r$.
S_C	present tag set, $S_C = S_R \bigcup S_O$.
c	cardinality of S_C, $c = \|S_C\|$, $c = r + o$.
S_U	universal set, $S_U = S_K \bigcup S_C$.
u	cardinality of S_U, $u = \|S_U\|$, $u = m + r + o$.
u_{max}	upper bound of u.
d	dynamic degree, given by $\frac{m}{r}$.
d_{max}	upper bound of d.
$\hat{r}$	estimated # of r.
α	required relative error, $\alpha \in (0, 1)$.
β	required confidence level, $\beta \in (0, 1)$.
f	size of sub-frame.
$\aleph$	# of sub-frames.
fc	frame counter.
sc	slot counter.
$H(\cdot)$	uniform hashing function.
R	random number.
$E(\cdot)$	expectation.
$D(\cdot)$	variance.
$F_1[\cdot]$	expected slot status vector.
$F_2[\cdot]$	observed slot status vector.
$\hat{u}_i$	estimated # of u after the i^{th} frame.
u_{max_i}	a tighter upper bound of u after the i^{th} frame.
$\hat{d}_i$	estimated # of d after the i^{th} frame.
d_{max_i}	a tighter upper bound of d after the i^{th} frame.
Z_β	the percentile of β that satisfies $P[-Z_\beta \leq W \leq Z_\beta] \geq \beta$, where W is a variable following standard normal distribution.

III. B-KC: Basic Key tag Counting Protocol

A. Communication Overview

This section presents the MAC layer communication mechanism of the *Basic Key tag Counting (B-KC)* protocol, which is based on the classical slotted Aloha algorithm. During the entire execution process of slotted Aloha-based protocols, all tags including those that have been identified are required to stay powered up to maintain the value of the *inventory* flag [31] [32]. Any intermittent loss of power at a tag will set its inventory flag back to 0, leading the tag to contend in the subsequent frame. Since long slotted frame will increase the risk of losing power at a tag, the frame size is typically no more than 512 [19] [23] [32]. In this paper, f is fixed to 512 for simplicity.

As exemplified in Fig. 2, the reader sequentially initializes $\aleph$ time frames to "load" the large number of tags, where each frame contains f slots. Specifically, the reader initializes an arbitrary frame with *frame counter* $fc \in [0, \aleph - 1]$ by broadcasting a request $\langle fc, \aleph, f, R \rangle$, in which R is a random number. Each tag calculates $H(ID, R) \mod \aleph$ to determine if it will participate in the current frame. If $H(ID, R) \mod \aleph$ is equal to the current *frame counter* fc, it will participate in the current frame (this can be implemented by the *SELECT* command of EPC C1G2 standard [31]). Note that, R does not change among all the sub-frames, and thus each tag will pseudo-randomly determine only one sub-frame to participate.

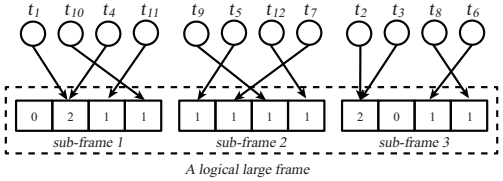

Fig. 2: The tags are "loaded" into multiple sub-frames, each sub-frame contains $f \leq 512$ slots.

The tags participating in the current frame will pick the sc^{th} slot, where $sc = H(ID, R) \mod f$. Each tag responds the 10-bit *checksum* [29] of its ID in the picked slot, which can be implemented by asserting *Truncate* bit in the *SELECT* command [31]. The reader needs to observes the status of slots by listening to the communication channel.

B. Estimation Protocol

This section investigates how to use the observed slot status to perform the key tag counting.

1) Overview of the Protocol Design: As illustrated in Fig. 2, the tags *logically* content for a large *logical* time frame that consists of $\aleph$ actual sub-frames, and each sub-frame contains f slots. Assuming the key tag set is exactly the same as the current tag set, the reader is able to predict the slot status vector $F_1[\cdot]$ because it knows the key tag set $S_K = \{x_1, x_2, ..., x_k\}$ and all the used parameters. However, the existence of missing key tags and the ordinary tags makes the observed slot status vector $F_2[\cdot]$ different from $F_1[\cdot]$. The proposed B-KC leverages the status change of the expected singleton slots (i.e., the singleton slot in $F_1[\cdot]$) to estimate the cardinality r of the remaining key tags.

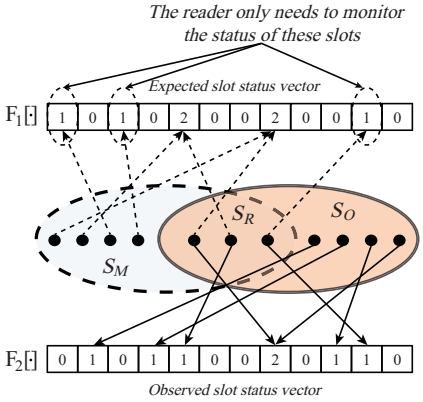

Fig. 3: The basic principle of the counting protocol.

2) Constructing the Vectors $F_1[\cdot]$ and $F_2[\cdot]$: The reader pseudo-randomly maps the key tags within S_K to a virtual frame F_1 with $\aleph f$ slots. Specifically, an arbitrary key tag, says $x_j \in S_K$ ($j \in [1, k]$), is mapped to the location of $H(ID_{x_j}, R) \mod \aleph f$ whose result follows a uniform distribution within $[0, \aleph f - 1]$, where ID_{x_j} is its ID information and R is a random number. In $F_1[\cdot]$, '0' in means no key tag is mapped to this location; '1' indicates only one key tag is mapped

to this location; '2' represents two or more key tags are mapped to this location. These three types of slots are called *expected empty slots*, *expected singleton slots*, and *expected collision slots*, respectively. The reader sequentially observes the $\aleph$ separate sub-frames (each contains f slots), and then constructs another vector $F_2[\cdot]$ with $\aleph f$ elements. If the sc^{th} slot in the fc^{th} sub-frame is an empty (singleton or collision) slot, $F_2[fc \times f + sc]$ is set to '0' ('1' or '2'). The two slots with the same location in $F_1[\cdot]$ and $F_2[\cdot]$ are called a *slot pair*. In our scheme, the reader needs to observe the expected singleton slots and record the numbers of the following two types of special slot pairs.

- $N_{1,0}$ is the number of slot pairs that satisfy: $F_1[z] = 1 \wedge F_2[z] = 0$.
- $N_{1,1}$ is the number of slot pairs that satisfy: $F_1[z] = 1 \wedge F_2[z] = 1 \wedge$ *the received checksum is the same as that of the expected tag*.

3) Proposing the Estimator of r: In the following, we theoretically present how to use the observed $N_{1,0}$ and $N_{1,1}$ to propose an accurate key tag counting estimator. First, we analyze the probabilistic properties behind the variables $N_{1,0}$ and $N_{1,1}$. Essentially, $N_{1,0}$ is equal to the number of missing tags in S_M that *exclusively* occupy slots. For an arbitrary missing tag, the probability that its picked slot is not selected by any other tags is denoted as $p_{1,0}$, which can be given as follows:

$$p_{1,0} = (1 - \frac{1}{\aleph f})^{u-1} \approx e^{-\frac{u}{\aleph f}} \tag{1}$$

Since $\aleph f$ is normally large enough, $N_{1,0}$ follows $Bernoulli(m, p_{1,0})$ distribution. And thus, the expectation $E(N_{1,0})$ and variance $D(N_{1,0})$ of $N_{1,0}$ are given as follows:

$$E(N_{1,0}) = m \times p_{1,0} = me^{-\frac{u}{\aleph f}} \tag{2}$$

$$\begin{aligned} D(N_{1,0}) &= m \times p_{1,0} \times (1 - p_{1,0}) \\ &= me^{-\frac{u}{\aleph f}}(1 - e^{-\frac{u}{\aleph f}}) \end{aligned} \tag{3}$$

Then, let us consider the variable $N_{1,1}$. *Two and only two* possible cases could contribute to $N_{1,1}$.

Case a: If a remaining key tag *exclusively* occupies a slot within the frame, $N_{1,1}$ will be increased by 1. We denote the number of this type of slot pairs as $N^a_{1,1}$. For an arbitrary remaining key tag, the probability that it exclusively occupies a slot is denoted as $p^a_{1,1}$, which can be given as follows:

$$p^a_{1,1} = (1 - \frac{1}{\aleph f})^{u-1} \approx e^{-\frac{u}{\aleph f}} \tag{4}$$

Since $N^a_{1,1}$ follows $Bernoulli(r, p^a_{1,1})$ distribution, the expectation $E(N^a_{1,1})$ and variance $D(N^a_{1,1})$ of $N^a_{1,1}$ are given as follows:

$$E(N^a_{1,1}) = r \times p^a_{1,1} = re^{-\frac{u}{\aleph f}} \tag{5}$$

$$\begin{aligned} D(N^a_{1,1}) &= r \times p^a_{1,1} \times (1 - p^a_{1,1}) \\ &= re^{-\frac{u}{\aleph f}}(1 - e^{-\frac{u}{\aleph f}}) \end{aligned} \tag{6}$$

Case b: If exactly a missing key tag as well as an ordinary tag pick a common slot, and their checksums are coincidentally the same, $N_{1,1}$ will be also increased by 1. And the number of this type of slot pairs is denoted as $N^b_{1,1}$. For an arbitrary missing key tag, the probability that it shares a common slot with only one ordinary tag and their 10-bit checksums are the same is denoted as $p^b_{1,1}$. We reasonably assume two arbitrary tags have the same 10-bit checksum with the probability $\frac{1}{2^{10}}$. Thus, $p^b_{1,1}$ can be given as follows:

$$\begin{aligned} p^b_{1,1} &= \binom{o}{1} \times \frac{1}{\aleph f} \times (1 - \frac{1}{\aleph f})^{u-2} \times \frac{1}{2^{10}} \\ &\approx \frac{oe^{-\frac{u}{\aleph f}}}{2^{10}\aleph f} \end{aligned} \tag{7}$$

Since $N^b_{1,1}$ follows $Bernoulli(m, p^b_{1,1})$ distribution, the expectation $E(N^b_{1,1})$ and variance $D(N^b_{1,1})$ of $N^b_{1,1}$ are given as follows:

$$E(N^b_{1,1}) = m \times p^b_{1,1} \quad = \frac{moe^{-\frac{u}{\aleph f}}}{2^{10}\aleph f} \tag{8}$$

$$\begin{aligned} D(N^b_{1,1}) =& m \times p^b_{1,1} \times (1 - p^b_{1,1}) \\ =& \frac{moe^{-\frac{u}{\aleph f}}}{2^{10}\aleph f}(1 - \frac{oe^{-\frac{u}{\aleph f}}}{2^{10}\aleph f}) \end{aligned} \tag{9}$$

Since $N_{1,1}$ consists of two parts: $N^a_{1,1}$ and $N^b_{1,1}$, $N_{1,1} = N^a_{1,1} + N^b_{1,1}$. The variables $N^a_{1,1}$ and $N^b_{1,1}$ are considered to be independent to each other, because $\aleph f$ is very large. Thus, we have $E(N_{1,1}) = E(N^a_{1,1}) + E(N^b_{1,1})$; and $D(N_{1,1}) = D(N^a_{1,1}) + D(N^b_{1,1})$. Comparing $E(N^a_{1,1})$ in Eq. (5) and $E(N^b_{1,1})$ in Eq. (8), we find that $E(N^b_{1,1})$ is so minor that it can be ignored. Similarly, compared with $D(N^a_{1,1})$, $D(N^b_{1,1})$ can also be ignored. We then have:

$$E(N_{1,1}) \approx E(N^a_{1,1}) = re^{-\frac{u}{\aleph f}} \tag{10}$$

$$D(N_{1,1}) \approx D(N^a_{1,1}) = re^{-\frac{u}{\aleph f}}(1 - e^{-\frac{u}{\aleph f}}) \tag{11}$$

According to Eqs. (2) and (10), we have:

$$r = \frac{k}{\frac{E(N_{1,0})}{E(N_{1,1})} + 1} \tag{12}$$

By substituting $N_{1,0}$ for $E(N_{1,0})$ and $N_{1,1}$ for $E(N_{1,1})$ in Eq. (12), we get the estimator of r as follows:

$$\hat{r} = \frac{k}{\frac{N_{1,0}}{N_{1,1}} + 1}, \tag{13}$$

where $N_{1,0}$ and $N_{1,1}$ are variables known from the observations, and $\hat{r}$ is the estimation result.

C. Investigating the Accuracy of B-KC

The following Theorem presents the expectation and variance of the estimator $\hat{r}$.

Theorem 1: When the frame size $\aleph f$ is large enough, $\hat{r}$ in Eq. (13) is an unbiased estimator of r. That is,

$$E(\hat{r}) = r \tag{14}$$

And the variance of the estimator $\hat{r}$ is as follows:

$$D(\hat{r}) = \frac{mr}{k}(e^{\frac{u}{\aleph f}} - 1), \tag{15}$$

where $k = m + r$ and $u = m + r + o$.

Proof: Since $\hat{r} = \frac{k}{\frac{N_{1,0}}{N_{1,1}}+1}$ Eq. (13) is a function of $N_{1,0}$ and $N_{1,1}$, it is denoted as $g(N_{1,0}, N_{1,1})$. We leverage Taylor series expansion [24] to get the expectation and variance of

r. In what follows, we present the Taylor series expansion of function $g(N_{1,0}, N_{1,1})$ around (θ_1, θ_2), where $\theta_1 = E(N_{1,0})$, $\theta_2 = E(N_{1,1})$.

$$\begin{aligned} & g(N_{1,0}, N_{1,1}) \\ \approx & g(\theta_1, \theta_2) + [(N_{1,0} - \theta_1)\frac{\partial g}{\partial N_{1,0}} + (N_{1,1} - \theta_2)\frac{\partial g}{\partial N_{1,1}}] \\ & + \frac{1}{2}[(N_{1,0} - \theta_1)^2 \frac{\partial^2 g}{\partial N_{1,0}^2} + 2(N_{1,0} - \theta_1)(N_{1,1} - \theta_2)\frac{\partial^2 g}{\partial N_{1,0}\partial N_{1,1}} \\ & + (N_{1,1} - \theta_2)^2 \frac{\partial^2 g}{\partial N_{1,1}^2}] \end{aligned} \tag{16}$$

Taking the expectation of both sides, we have:

$$\begin{aligned} & E[g(N_{1,0}, N_{1,1})] \\ = & g(\theta_1, \theta_2) + \frac{1}{2}[D(N_{1,0})\frac{\partial^2 g}{\partial N_{1,0}^2} + 2Cov(N_{1,0}, N_{1,1})\frac{\partial^2 g}{\partial N_{1,0}\partial N_{1,1}} \\ & + D(N_{1,1})\frac{\partial^2 g}{\partial N_{1,1}^2}] \\ = & g(\theta_1, \theta_2) + \frac{1}{2}[D(N_{1,0})\frac{\partial^2 g}{\partial N_{1,0}^2} + D(N_{1,1})\frac{\partial^2 g}{\partial N_{1,1}^2}] \end{aligned} \tag{17}$$

In Eq. (17), $N_{1,0}$ and $N_{1,1}$ are independent to each other when considering $\aleph f$ is large enough. Thus, $Cov(N_{1,0}, N_{1,1})$ is simplified to 0. As required in Eq. (17), the second-order partial derivatives of function $g(N_{1,0}, N_{1,1})$ are calculated as follows.

$$\begin{cases} \frac{\partial^2 g(N_{1,0}, N_{1,1})}{\partial N_{1,0}^2} \mid_{N_{1,1}=\theta_2}^{N_{1,0}=\theta_1} = \frac{2k\theta_2}{(\theta_1 + \theta_2)^3} \\ \frac{\partial^2 g(N_{1,0}, N_{1,1})}{\partial N_{1,1}^2} \mid_{N_{1,1}=\theta_2}^{N_{1,0}=\theta_1} = \frac{-2k\theta_1}{(\theta_1 + \theta_2)^3} \end{cases}$$

Putting the above values into Eq. (17), and replacing θ_1 by $E(N_{1,0})$, θ_2 by $E(N_{1,1})$, we then have:

$$\begin{aligned} & E[g(N_{1,0}, N_{1,1})] \\ = & g[E(N_{1,0}), E(N_{1,1})] + N[\frac{E(N_{1,1})D(N_{1,0}) - E(N_{1,0})D(N_{1,1})}{[E(N_{1,0}) + E(N1,1)]^3}] \end{aligned} \tag{18}$$

Combining the expectations and variances of $N_{1,0}$ and $N_{1,1}$ in Eqs. (2), (3), (10), and (11) into Eq. (18), we have:

$$\begin{aligned} E(\hat{r}) &= E[g(N_{1,0}, N_{1,1})] \\ &= g[E(N_{1,0}), E(N_{1,1})] \\ &= \frac{k}{\frac{E(N_{1,0})}{E(N_{1,1})} + 1} \\ &= r \end{aligned} \tag{19}$$

Eq. (19) indicates that $\hat{r}$ is an unbiased estimator of r. The variance $D(\hat{r})$ of $\hat{r}$ is calculated as follows:

$$\begin{aligned} D(\hat{r}) &= E[\hat{r} - E(\hat{r})]^2 \\ &= E[g(N_{1,0}, N_{1,1}) - r]^2 \end{aligned} \tag{20}$$

We use the first-order Taylor series expansion of $g(N_{1,0}, N_{1,1})$ to substitute it in Eq. (20). Thus, we have:

$$\begin{aligned} D(\hat{r}) =& E[g(N_{1,0}, N_{1,1}) - r]^2 \\ =& E[(N_{1,0} - \theta_1)\frac{\partial g}{\partial N_{1,0}} + (N_{1,1} - \theta_2)\frac{\partial g}{\partial N_{1,1}}]^2 \\ =& E[(N_{1,0} - \theta_1)^2(\frac{\partial g}{\partial N_{1,0}})^2 + (N_{1,1} - \theta_2)^2(\frac{\partial g}{\partial N_{1,1}})^2 \\ & + 2(N_{1,0} - \theta_1)(N_{1,1} - \theta_2)(\frac{\partial g}{\partial N_{1,0}})(\frac{\partial g}{\partial N_{1,1}})] \\ =& D(N_{1,0})(\frac{\partial g}{\partial N_{1,0}})^2 + D(N_{1,1})(\frac{\partial g}{\partial N_{1,1}})^2 \end{aligned} \tag{21}$$

As required in Eq. (21), the first-order partial derivatives of function $g(N_{1,0}, N_{1,1})$ are calculated as follows.

$$\begin{cases} \frac{\partial g(N_{1,0}, N_{1,1})}{\partial N_{1,0}} \mid_{N_{1,1}=\theta_2}^{N_{1,0}=\theta_1} = \frac{-k\theta_2}{(\theta_1 + \theta_2)^2} \\ \frac{\partial g(N_{1,0}, N_{1,1})}{\partial N_{1,1}} \mid_{N_{1,1}=\theta_2}^{N_{1,0}=\theta_1} = \frac{k\theta_1}{(\theta_1 + \theta_2)^2} \end{cases}$$

Putting the above values into Eq. (21) and replacing θ_1 by $E(N_{1,0})$, θ_2 by $E(N_{1,1})$, we then have:

$$D(\hat{r}) = \frac{k^2[E^2(N_{1,1})D(N_{1,0}) + E^2(N_{1,0})D(N_{1,1})]}{[E(N_{1,0}) + E(N_{1,1})]^4} \tag{22}$$

Combining the expectations and variances of $N_{1,0}$ and $N_{1,1}$ in Eqs. (2), (3), (10), and (11) into Eq. (23), we have:

$$D(\hat{r}) = \frac{mr}{k}(e^{\frac{u}{\aleph f}} - 1), \tag{23}$$

that is, Eq. (15) is approved.

■

Considering the required accuracy of the proposed estimator, one may ask the following question.

Question 1. *How many sub-frames are adequate to make B-KC meet the required (α, β) accuracy, i.e., $P\{|\hat{r} - r| \leq \alpha \cdot r\} \geq \beta$.*

We propose the following Theorem to give the answer to *Question 1*.

Theorem 2: If the number $\aleph$ of sub-frames is not less than $u/[f \ln(\frac{k\alpha^2}{dZ_\beta^2} + 1)]$, the estimation result $\hat{r}$ will meet the predefined accuracy (α, β), that is, $P\{|\hat{r} - r| \leq \alpha \cdot r\} \geq \beta$.

Proof: According to the central limit theorem [25], we have that $W = \frac{\hat{r} - E(\hat{r})}{\sqrt{D(\hat{r})}}$ satisfies the standard normal distribution. We can find a percentile Z_β of β such that $P\{-Z_\beta \leq W \leq Z_\beta\} \geq \beta$. For example, if $\beta = 95\%$ then $Z_\beta = 1.96$. The required estimation accuracy can be rewritten as follows:

$$\begin{aligned} & P\{|\hat{r} - r| \leq \alpha \cdot r\} \\ = & P\{(1 - \alpha)r \leq \hat{r} \leq (1 + \alpha)r\} \\ = & P\{\frac{(1 - \alpha)r - E(\hat{r})}{\sqrt{D(\hat{r})}} \leq \frac{\hat{r} - E(\hat{r})}{\sqrt{D(\hat{r})}} \leq \frac{(1 + \alpha)r - E(\hat{r})}{\sqrt{D(\hat{r})}}\} \end{aligned} \tag{24}$$

According to Eq. (24), if we have the following inequalities:

$$\begin{cases} \frac{(1 - \alpha)r - E(\hat{r})}{\sqrt{D(\hat{r})}} \leq -Z_\beta \\ \frac{(1 + \alpha)r - E(\hat{r})}{\sqrt{D(\hat{r})}} \geq Z_\beta, \end{cases}$$

we can guarantee $P\{|\hat{r}-r| \leq \alpha \cdot r\} \geq \beta$. Substituting $E(\hat{r}) = r$ and $D(\hat{r}) = \frac{mr}{k}(e^{\frac{u}{\aleph f}} - 1)$ into the above inequalities and solving them, we have:

$$\aleph \geq u/[f \ln(\frac{k\alpha^2}{dZ_\beta^2} + 1)], \tag{25}$$

where $d = \frac{m}{r}$ is used to describe the *dynamic degree* of the key tag set. ■

Theorem 2 presents how to configure $\aleph$ to produce an (α, β) estimate of r. However, u and d is not known in priori. Because the minimum $\aleph$ is a monotonically increasing function with respect to u and d, we first use $u = u_{max}$ and $d = d_{max}$ to calculate $\aleph$ such that the (α, β) accuracy can be always satisfied for any actual u and d.

D. Skipping the Expected Empty/Collision Slots

According to the estimator in Eq. (13), the reader only needs to monitor the status of the *expected singleton slots*. In other words, the expected empty slots as well as the expected collision slots are not used at all, and their execution wastes a large amount of time. Exploiting the methods used in [33] [34], the expected empty slots and collision slots can be directly skipped without execution. The *slot skipping* method is described in the following. Before each sub-frame with f slots, the reader constructs a bitmap with f bits, in which '1s' mean the expected singleton slots that need to be executed and '0s' represent the expected empty/collision slots. For a certain tag, the bit corresponding to its picked slot is referred to as the *representative bit*. When receiving the bitmap, a tag counts the number λ of '1s' proceeding its representative bit. If the representative bit of a tag is '1', it will respond in the $(\lambda+1)^{th}$ slots . In contrary, if a tag finds its presentative bit is '0', which means it picks an expected collision slot, it will not respond at all. As a result of the above procedures, only expected singleton slots are executed, and a large number of expected empty/collision slots that are not used are skipped.

E. Time Cost of B-KC

In the following, let us consider the execution time of the proposed B-KC which consists of three parts. **Transmission of Initial Parameters.** For an arbitrary sub-frame, a tag slot t_{tag} is adequate to broadcast the initialization parameters $\langle fc, \aleph, f, R\rangle$. **Transmission of f-bit Bitmap.** The bitmap is divided into 96-bit segments to to be transmitted in $\lceil \frac{f}{96} \rceil$ tag slots. **Execution of the Expected Singleton Slots.** An arbitrary slot in this sub-frame has the following probability to be an expected singleton slot.

$$\begin{aligned} p_{1,*} &= \binom{k}{1} \times \frac{1}{\aleph} \times \frac{1}{f} \times (1 - \frac{1}{\aleph} \times \frac{1}{f})^{k-1} \\ &\approx \frac{ke^{-\frac{k}{\aleph f}}}{\aleph f} \end{aligned} \tag{26}$$

And then, the number of expected singleton slots that need to be executed in this sub-frame is $f \times p_{1,*} = \frac{k}{\aleph} e^{-\frac{k}{\aleph f}}$. Combining the above three parts of time, the time cost of this sub-frame is $t_{tag} + \lceil \frac{f}{96} \rceil t_{tag} + \frac{k}{\aleph} e^{-\frac{k}{\aleph f}} t_{long}$. For $\aleph$ sub-frames in total, the whole execution time of B-KC denoted as T_B is given as follows:

$$\begin{aligned} T_B &= \aleph \times (t_{tag} + \lceil \frac{f}{96} \rceil t_{tag} + \frac{k}{\aleph} e^{-\frac{k}{\aleph f}} t_{long}) \\ &= \aleph t_{tag} + \aleph \lceil \frac{f}{96} \rceil t_{tag} + k e^{-\frac{k}{\aleph f}} t_{long} \end{aligned} \tag{27}$$

IV. S-KC: Sampling-based Key tag Counting Protocol

of the most important requirements of a tag estimation scheme is *scalability*, i.e., the estimation time needs to be scalable to large population sizes [19]. However, the numerical results in Fig. 4 reveal that the execution time of B-KC increases sharply with the increase of u_{max}. Therefore, the scalability of B-KC needs to be further improved. Based on B-KC, this section first exploits the *sampling idea* [35] [36] to propose the *Sampling-based Key tag Counting (S-KC)* protocol. Actually, B-KC is a special case ($p = 1$) of S-KC. With the involved parameter p, we also propose the theoretical analysis to investigate the parameter settings of S-KC. Via numerical results, we observe that even though S-KC outperforms B-KC, its time-efficiency is still far from the ideal case. Then, we propose a tactic named *early termination* to bridge the gap between the performance of S-KC and the ideal case.

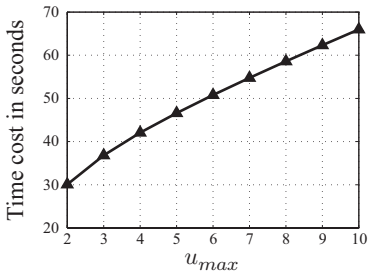

Fig. 4: Evaluating the execution time of B-KC against u_{max}. $\alpha = 5\%$, $\beta = 95\%$, and k is fixed to $50,000$; $d_{max} = 8$; u_{max} varies from $200,000$ to $1,000,000$.

A. Overview of S-KC

1) Scheme Overview: In the new estimation scheme, each tag participates in the estimation process with a probability of p, namely the *sampling probability*. A tiny sampling method suitable for the RFID devices was proposed in [35]. The reader broadcasts a request $\langle R', l\rangle$, where R' is a new random number and the integer l is calculated by $\lceil p \times L \rceil$, in which L is a sufficiently large constant pre-configured in the tags during the manufacturing process. Using the received random seed and its ID, each tag calculates $H(ID, R') \mod L$ whose result follows a uniform distribution within $[0, L)$. If the hashing result is less than the received parameter l, the tag will participate in the estimation process; otherwise, it will not. As a result, each tag participates in the estimation process with a probability of p. The rest of procedures is the same as that of B-KC.

2) Proposing the New Estimator of r: We still use the observations of $N_{1,0}$ and $N_{1,1}$ to estimate r. To differentiate the new analytical procedures from those in the last section, we introduce two new notations but with the same physical meaning as the original ones, $N'_{1,0}$ and $N'_{1,1}$. Similar to the theoretical analysis in Section III, the expectations and variances of $N'_{1,0}$ and $N'_{1,1}$ are given as follows:

$$\begin{aligned} E(N'_{1,0}) &= mpe^{-\frac{up}{\aleph f}} \\ D(N'_{1,0}) &= m(pe^{-\frac{up}{\aleph f}})(1-pe^{-\frac{up}{\aleph f}}) \\ E(N'_{1,1}) &= rpe^{-\frac{up}{\aleph f}} \\ D(N'_{1,1}) &= r(pe^{-\frac{up}{\aleph f}})(1-pe^{-\frac{up}{\aleph f}}) \end{aligned} \tag{28}$$

According to Eq. (28), the new estimator can be given as follows:

$$\hat{r}' = \frac{k}{\frac{N'_{1,0}}{N'_{1,1}}+1} \tag{29}$$

B. Investigating the Accuracy of S-KC

In the following, Theorem 3 gives expectation and variance of the new estimator $\hat{r}'$.

Theorem 3: When the frame size $\aleph f$ is large enough, $\hat{r}'$ is an approximately unbiased estimator of r. That is,

$$E(\hat{r}') = r \tag{30}$$

And the variance of the estimator $\hat{r}'$ is as follows:

$$D(\hat{r}') = \frac{mr}{k}(\frac{1}{p}e^{\frac{up}{\aleph f}} - 1) \tag{31}$$

where $k = m + r$ and $u = m + r + o$.

Proof: Using equations in Eqs. (28) and (29), this theorem can be similarly deduced from proof of Theorem 1. ■

Then, we propose the following Theorem to investigate how many sub-frames are adequate to make S-KC meet the required (α, β) accuracy.

Theorem 4: With a fixed sampling probability $p \in (\frac{Z_\beta^2 d}{k\alpha^2+Z_\beta^2 d}, 1]$, if the number $\aleph$ of sub-frames is not less than $up/\{f\ln[(\frac{k\alpha^2}{Z_\beta^2 d}+1)p]\}$, the estimation result $\hat{r}'$ will meet the predefined accuracy (α, β), that is, $P\{|\hat{r}'-r| \le \alpha \cdot r\} \ge \beta$.

Proof: According to equations in Eqs. (30) and (31), we need to guarantee $\aleph \ge up/\{f\ln[(\frac{k\alpha^2}{Z_\beta^2 d}+1)p]\}$ in order to meet the predefined accuracy (α, β). This can be deduced from proof of Theorem 2. Please note that, not all $p \in (0, 1]$ can be used. If p is too small, the denominator $f\ln[(\frac{k\alpha^2}{Z_\beta^2 d}+1)p]$ will become negative. By solving $\ln[(\frac{k\alpha^2}{Z_\beta^2 d}+1)p] > 0$, we get the ranges of the sampling probability p as $(\frac{Z_\beta^2 d}{k\alpha^2+Z_\beta^2 d}, 1]$. ■

Clearly, the expression of $\aleph$ proposed in Theorem 4 is still an increasing function against u and d. Thus, $\aleph$ should be calculated by $u = u_{max}$ and $d = d_{max}$ so as to accommodate any actual u and d.

C. Time Cost of S-KC

Similar with the analysis in Section III-E, the whole execution time of S-KC, denoted as T_S, is given as follows:

$$T_S = \aleph t_{tag} + \aleph\lceil\frac{f}{96}\rceil t_{tag} + kpe^{-\frac{kp}{\aleph f}} t_{long} \tag{32}$$

As illustrated in Fig. 5, the configuration of sampling probability significantly affects the performance of S-KC. We can use an offline Algorithm 1 to find the optimal sampling probability p_{op} before performing the estimation.

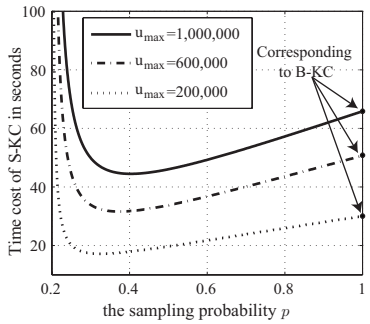

Fig. 5: Evaluating the execution time of S-KC against p. $\alpha = 5\%$, $\beta = 95\%$, and k is fixed to $50,000$; $d_{max} = 8$; p varies from 0.2 to 1.

Algorithm 1: Find the optimal sampling probability p_{op} and the corresponding number $\aleph_{op}$ of sub-frames as the inputs of S-KC.

Input: k, u_{max}, d_{max}, α, Z_β, f.
Output: the optimal sampling probability p_{op} and sub-frame number $\aleph_{op}$.

```
Time = +∞;
p_min = Z_β^2 d_max / (kα^2 + Z_β^2 d_max);
δ = 0.01;
for each p ∈ (p_min, 1] with step δ do
    ℵ = u_max p / {f ln[(kα^2 / (Z_β^2 d_max) + 1) p]};
    T = ℵ t_tag + ℵ⌈f/96⌉ × t_tag + kpe^(−kp/ℵf) × t_long;
    if T < Time then
        p_op = p;
        ℵ_op = ℵ;
        Time = T;
    end if
end for
return p_op and ℵ_op;
```

D. Early Termination

1) Motivation: Because the actual u and d are not known in prior, we have to use the extreme values (i.e., u_{max}, d_{max}) of them to calculate the minimum sub-frame number $\aleph$. However, the numerical results illustrated in Fig. 6 reveal the big performance **gap** between the time by using u_{max} and

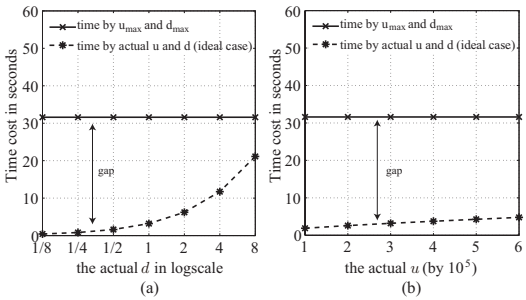

Fig. 6: Performance gap: time by u_{max} and d_{max} vs. time by actual u and d. $k = 50,000$, $\alpha = 5\%$, $\beta = 95\%$, $u_{max} = 600,000$, $d_{max} = 8$. (a) the actual u is fixed to $300,000$, the actual d varies from $\frac{1}{8}$ to 8 in logscale; (b) the actual d is fixed to 1, the actual u varies from $100,000$ to $600,000$.

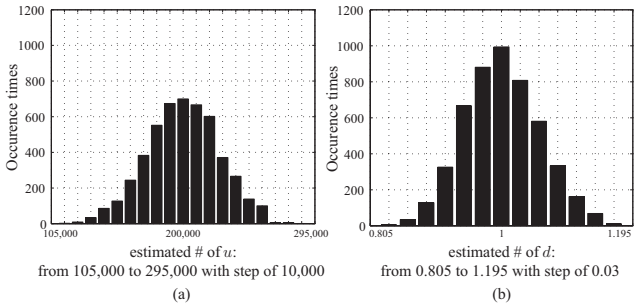

Fig. 7: Investigating the distribution of $\hat{u}_i$ and $\hat{d}_i$. $t = 50,000$, $u = 200,000$, $d = 1$. (a) recording the estimate $\hat{u}$ after 50^{th} sub-frame with $5,000$ independent trials; (b) recording the estimate $\hat{d}$ after 50^{th} sub-frame with $5,000$ independent trials.

d_{max} to calculate $\aleph$ and the time by using the actual u and d to calculate $\aleph$. An immediate question is as follows.

Question 2. *How to make the performance of S-KC approach its ideal case (i.e., assuming u and d are known in prior).*

To answer *Question 2*, this section proposes a tactic named **early termination** to bridge this gap. Specifically, at the very beginning, we configure the parameters p and $\aleph$ based on u_{max} and d_{max}. After an arbitrary sub-frame $i \in [0, \aleph - 1]$, we leverage the observation of the first $i+1$ sub-frames that have already been executed to give tighter upper bounds u_{max_i} on u and d_{max_i} on d. Based on the new u_{max_i} and d_{max_i}, the backend server determines if the current estimation result of r has already met the required accuracy (α, β). If so, the reader will terminate the execution right now, otherwise, the next sub-frame will be executed.

2) Giving the Tighter Bounds of u and d: According to Eq. (28), we first leverage the observed $N'^i_{1,0}$ and $N'^i_{1,1}$ after the i^{th} sub-frame to approximate u and d as follows.

$$\begin{aligned} \hat{u}_i &= -\frac{\aleph f}{p} \ln(\frac{N'^i_{1,0} + N'^i_{1,1}}{kp_i}) \\ \hat{d}_i &= \frac{N'^i_{1,0}}{N'^i_{1,1}}, \end{aligned} \tag{33}$$

where the actual sampling probability p_i is equal to $\frac{(i+1)p}{\aleph}$. Similar with Section III-C, we get the expectation and deviation of $\hat{u}_i$ and $\hat{d}_i$ as follows, respectively.

$$\begin{aligned} E(\hat{u}_i) &= u \\ D(\hat{u}_i) &= \frac{\aleph^2 f^2}{kp^2}(\frac{e^{\frac{up}{\aleph f}}}{p_i} - 1) \\ E(\hat{d}_i) &= d \\ D(\hat{d}_i) &= \frac{d(d+1)^2}{k} \times (\frac{e^{\frac{up}{\aleph f}}}{p_i} - 1) \end{aligned} \tag{34}$$

The well-known *three-sigma rule* [26] indicates that: *if a variable V follows the normal distribution, then it can differ from its expectation $E(V)$ by a quantity exceeding $3\sqrt{D(V)}$ with a probability no more than 0.3%.* The simulation results in Fig. 7 reveal that both $\hat{u}_i$ and $\hat{d}_i$ approximately follow the normal distribution. Hence, we have the following inequalities.

$$\begin{aligned} &P[E(\hat{u}_i) - 3\sqrt{D(\hat{u}_i)} < \hat{u}_i < E(\hat{u}_i) + 3\sqrt{D(\hat{u}_i)}] > 99.7\% \\ &P[E(\hat{d}_i) - 3\sqrt{D(\hat{d}_i)} < \hat{d}_i < E(d_i) + 3\sqrt{D(\hat{d}_i)}] > 99.7\% \end{aligned} \tag{35}$$

According to Eqs. (34) (35), we can get the new upper bounds of u and d as follows.

$$\begin{aligned} u_{max\_i} &= \hat{u}_i + \frac{3\aleph f}{p}\sqrt{\frac{1}{k}(\frac{e^{\frac{\hat{u}_i p}{\aleph f}}}{p_i} - 1)} \\ d_{max\_i} &= \hat{d}_i + 3(\hat{d}_i + 1)\sqrt{\frac{\hat{d}_i}{k}(\frac{e^{\frac{\hat{u}_i p}{\aleph f}}}{p_i} - 1)}, \end{aligned} \tag{36}$$

where $\hat{u}_i$ and $\hat{d}_i$ are the temporary estimation results got from Eq. (33).

3) The Conditions of Early Termination: According to Theorem 4, if the following two conditions are satisfied simultaneously, the required (α, β) estimation accuracy can be guaranteed. Then, the estimation process terminates.

$$\begin{aligned} p_i &> \frac{Z^2_\beta d_{max\_i}}{k\alpha^2 + Z^2_\beta d_{max\_i}} \\ (i+1) &\geq u_{max\_i} p_i / \{f \ln[(\frac{k\alpha^2}{Z^2_\beta d_{max\_i}} + 1)p_i]\} \end{aligned} \tag{37}$$

As illustrated in Fig. 8, the proposed early termination tactic can well bridge the performance gap discussed above, and thus makes the performance of S-KC very close to the ideal case.

E. Discussion on the Multi-reader Case

In a large scale application scenario, a single reader is usually not able to cover all the tags due to the limited communication ranges of RFID tags. In the following, we discuss how to extend the proposed S-KC to the multi-reader scenarios. Because of the overlapping region, new types of signal collisions such as reader-reader collisions and reader-tag collisions may arise [37], which has attracted much attention from research community [37] [38] [39]. Discussing these new types of collisions is beyond the scope this paper. Here, we only consider the tag-tag collisions, then assume the backend server could well synchronize the readers [19] [16]. Logically,

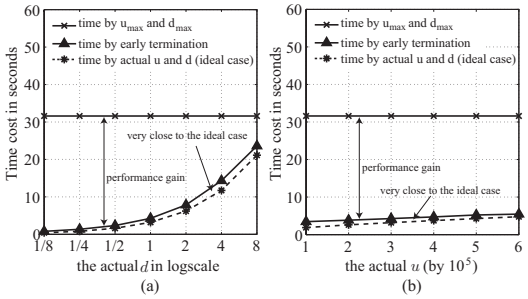

Fig. 8: Performance improvement benefiting from the tactic of early termination: $k = 50,000$, $\alpha = 5\%$, $\beta = 95\%$, $u_{max} = 600,000$, $d_{max} = 8$. (a) the actual u is fixed to $300,000$, the actual d varies from $\frac{1}{8}$ to 8 in logscale; (b) the actual d is fixed to 1, the actual u varies from $100,000$ to $600,000$.

the readers just paly the role of repeaters between the backend server and tags. Specifically, all the used parameters such as fc, f, $\aleph$, R are generated by the backend server and delivered to all the readers. The readers then query the tags by these global parameters. After each slot, the reader sends the received information (a single checksum or a collsion) to the backend server. The backend server constructs the global slot status vector based on the following rules: (1) this slot is empty if and only if (iff) all the readers detect an empty slot; (2) this slot is singleton iff a single reader returns a checksum or multiple readers return the same checksum; (3) otherwise, this slot is a collision slot. By comparing the expected slot status vector and the constructed actual slot vector, the backend server can perform the key tag counting process as what we have described before.

V. Performance Evaluation

In this section, we first conducted simulations to evaluate the performance of S-KC in large scale RFID system that consists of thousands of tags. Then, we implement a prototype of S-KC to evaluate its practical deployability.

A. Simulation

The simulators were implemented via MATLAB on a ThinkPad X230 desktop with an Intel i5 3230M CPU and 8G RAM. In the following, we first conduct a comparison on execution time between S-KC and prior schemes: CATS [11], ITSP [12], ZDE [22] and INC [23]. Note that, because the identification-based protocols are far from efficiency, we do not compare the proposed S-KC with them. Compared with the delay of wireless data transmission, the time consumed by computing on both the reader side and the tag side is so minor, and thus is neglected. Therefore, we only consider the time consumed by the wireless communications between the reader and the tags. Moreover, the same as the literature [11] [12] [22] [23], we consider a error-free communication channel. Then, we conduct experiments to show that S-KC indeed achieves the required estimation accuracy. Each simulation is conducted for 1000 times and we record the average results.

1) Execution time: In this section, we conduct simulations to evaluate the time-efficiency of the proposed S-KC. CATS and ITSP need the value of c (i.e., $|S_C|$) to optimize the parameter settings. Then, Zheng *et al.* proposed a light-weight scheme to roughly estimate c thereby providing input for CATS [11]. Chen *et al.* directly borrowed the efficient cardinality estimation protocol ART to estimate c [12]. To their favor, we do not take these time cost into account and configure their parameters using the actual c.

Investigating the Impact of u. In the simulations corresponding to Fig. 9, we aim at investigating the impact of u on the execution time needed by each scheme. Specifically, u_{max} is fixed to $600,000$, which is large enough for common applications. The cardinality k of key tags is set to $50,000$. The d_{max} is set to 8, and the actual d is configured to 1. The actual u varies from $100,000$ to $500,000$. We make two main observations from the results shown in Fig. 9 (a) and (b). First, the performance of CATS, ITSP and the proposed S-KC is not sensitive to u, whereas, the execution time of ZDE and INC increase linearly with respect to u. And the proposed S-KC is faster than than all prior schemes in all these configurations. For example, as illustrated in Fig. 9 (b), when $u = 500,000$, the execution time of CATS, ITSP, ZDE is 151.1s, 108.7s, 68.8s, respectively. Note that, the execution time of INC exceeds the bounds of the Fig. 9 (b) when $u = 500,000$. And the execution time of S-KC is just 5.3s, which represents 28.5 times faster than CATS, 20.5 times faster than ITSP, 13 times faster than ZDE. Second, by comparing Fig. 9 (a) and (b), we observe that the higher the required estimation accuracy is, the longer the execution time is, which holds on for each scheme.

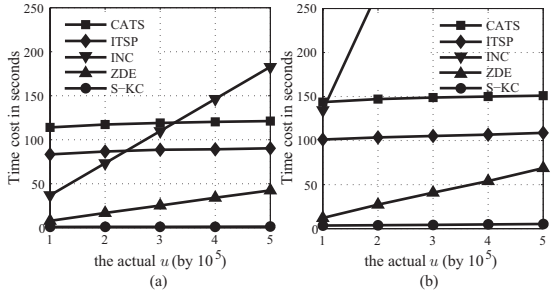

Fig. 9: Time vs. u: $k = 50,000$, $d_{max} = 8$, $d = 1$, $u_{max} = 600,000$, u varies from $100,000$ to $500,000$. (a) $\alpha = 10\%$, $\beta = 90\%$; (b) $\alpha = 5\%$, $\beta = 95\%$.

Investigating the Impact of k. In the simulations corresponding to Fig. 10, we aim at investigating the impact of k on the execution time needed by each scheme. Specifically, u_{max} is still fixed to $600,000$, and the actual u is set to $300,000$. The d_{max} is fixed to 8, and the actual d is set to 1. The cardinality k of the key tags varies from $30,000$ to $70,000$. We make two main observations from the results shown in Fig. 10 (a) and (b). First, the execution time of CATS and ITSP increases linearly with respect to k. In contrary, the execution time of ZDE, INC and our S-KC decreases

with respect to k. The underlying reason is that a larger k increases the ratio of $\frac{r}{u}$, which facilitates the estimation of r. Second, the proposed S-KC persistently outperforms the prior schemes with different k, which reveals its good scalability. As illustrated in Fig. 10 (b), when $t = 70,000$, the execution time of CATS, ITSP, ZDE is 206.4s, 143.9s, and 35.1s, respectively. Note that, because the execution time of INC *exceeds* the bounds of the Fig. 10 (b), the corresponding line does not appear. The execution time of S-KC is just 3.6s, which represents 57.3 times faster than CATS, 40 times faster than ITSP, 9.8 times faster than ZDE.

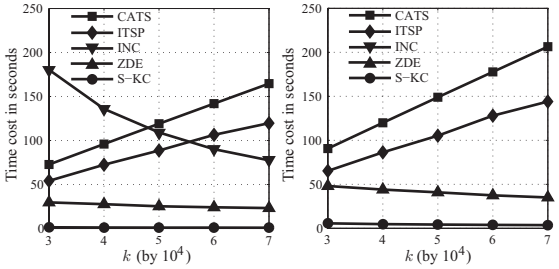

Fig. 10: Time vs. k: $d_{max} = 8$, $d = 1$, $u_{max} = 600,000$, $u = 300,000$, k varies from $30,000$ to $70,000$. (a) $\alpha = 10\%$, $\beta = 90\%$; (b) $\alpha = 5\%$, $\beta = 95\%$.

Investigating the Impact of d. In the simulations corresponding to Fig. 11, we aim at investigating the impact of d on the execution time needed by each scheme. Specifically, u_{max} is still fixed to $600,000$, and the actual u is set to $300,000$. The cardinality k of key tags is set to $50,000$. The upper bound d_{max} of d is configured to 8, and the actual d varies from $\frac{1}{4}$ to 4 in log-scale. According to Fig. 11 (a) and (b), we observe that the proposed S-KC is significantly faster than all prior schemes. As illustrated in Fig. 11 (b), when $d = 4$, the execution time of CATS, ITSP, ZDE is 207.9s, 72.7s, 109.9s, respectively. Again, the line corresponding to INC does not appear in Fig. 11 (b) because it exceeds the bounds of figure. And the execution time of S-KC is just 14.5s, which represents 13.3 times faster than CATS, 5 times faster than ITSP, and 7.6 times faster than ZDE.

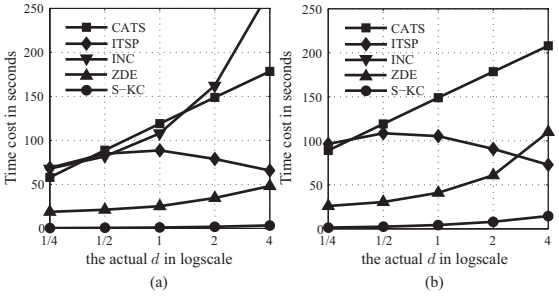

Fig. 11: Time vs. d: $k = 50,000$, $u_{max} = 600,000$, $u = 300,000$, $d_{max} = 8$, d varies from $\frac{1}{4}$ to 4 in logscale. (a) $\alpha = 10\%$, $\beta = 90\%$; (b) $\alpha = 5\%$, $\beta = 95\%$.

2) Actual Accuracy: The parameters (α, β) given as input of S-KC indicate the *required accuracy*. The estimation accuracy that an estimation scheme achieves is referred to as its *actual accuracy (or actual reliability)*. The actual accuracy should always be greater than or equal to the required accuracy [19]. Hence, this section conducts simulations to evaluate the actual accuracy of the proposed S-KC. Specifically, for each parameter setting, we conducted 1000 independent simulations. In an arbitrary simulation, if the estimation result $\hat{r}$ is within $[r(1-\alpha), r(1+\alpha)]$, we refer to it as a *success estimation*. We record the *success times* among 1000 times. We use $\frac{success\ times}{1000}$ to measure the *actual accuracy*. The simulation results shown in Figures 12, 13 and 14 demonstrate that the proposed S-KC always achieves the required accuracy.

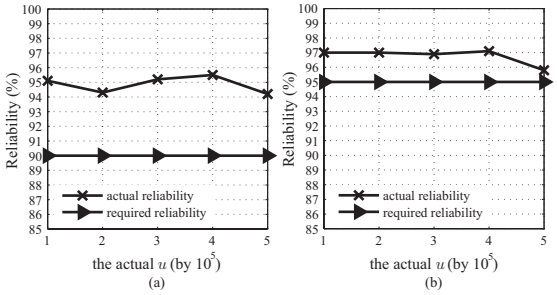

Fig. 12: Actual accuracy vs. u: $k = 50,000$, $d_{max} = 8$, $d = 1$, $u_{max} = 600,000$, u varies from $100,000$ to $500,000$. (a) $\alpha = 10\%$, $\beta = 90\%$; (b) $\alpha = 5\%$, $\beta = 95\%$.

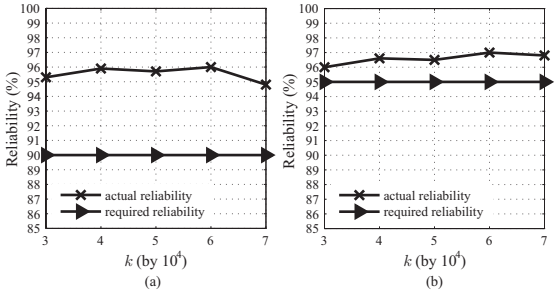

Fig. 13: Actual accuracy vs. k: $d_{max} = 8$, $d = 1$, $u_{max} = 600,000$, $u = 300,000$, k varies from $30,000$ to $70,000$. (a) $\alpha = 10\%$, $\beta = 90\%$; (b) $\alpha = 5\%$, $\beta = 95\%$.

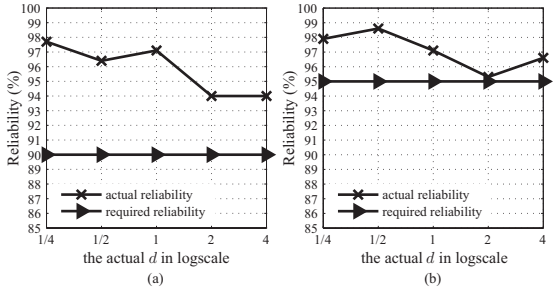

Fig. 14: Actual accuracy vs. d: $k = 50,000$, $u_{max} = 600,000$, $u = 300,000$, $d_{max} = 8$, d varies from $\frac{1}{4}$ to 4 in logscale. (a) $\alpha = 10\%$, $\beta = 90\%$; (b) $\alpha = 5\%$, $\beta = 95\%$.

B. Prototype Implementation

We use the highly integrated ultra low power 2.4GHz RF System-on-Chip (SoC) nRF24LE1 [40] to implement a prototype of our S-KC, which is shown in Fig. 15. nRF24LE1 includes a 2.4GHz RF transceiver core, an 8-bit CPU, and embedded Flash memory. The computer and the reader communicate via RS232 serial port. The tags are active and powered by button cells (3V). The prototype also includes a simple user interface on the computer side, by which the end users can configure the required estimation accuracy and get the estimation result. The implemented RFID system includes one RFID reader and 20 RFID tags. The specified key tag list contains 20 potential IDs. The present key tags (i.e., the tags in S_R) are fixed to 10. We conducted 100 independent experiments. As shown in Fig. 16, 95 estimation results among 100 simulations meet the predefined estimation accuracy ($\alpha = 0.1$), which demonstrates the correctness of the implemented prototype.

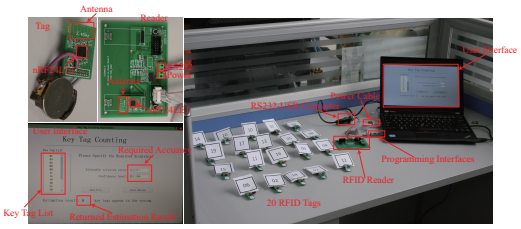

Fig. 15: The implemented prototype of our S-KC.

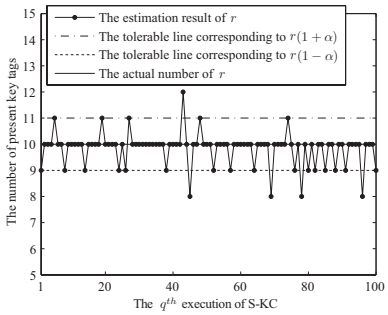

Fig. 16: Outputs of the implemented prototype. $\alpha = 10\%$, $\beta = 90\%$. The actual r is fixed to 10.

VI. Related Work

In RFID-enabled applications, one of the most fundamental tasks is *tag identification* that aims at identifying all the IDs of tags within the interrogation ranges of a reader. The identification protocols are generally classified into two categories: Aloha-based protocols [5] [6] [7] and Tree-based protocols [8] [9] [10] [32]. In ALOHA-based identification protocols, the reader queries the tags and periodically broadcasts synchronization signals to create a slotted time frame. Upon receiving such a request, each tag randomly picks a slot in the frame to relay its ID information. If a tag *exclusively* occupies a slot, its ID can be received by the reader. In contrary, if it shares a common slot with other tags, then its ID cannot be received by the reader due to the signal collision, and therefore retransmission is required [19]. The tree-based identification protocol is a recursive depth-first searching algorithm performed by the reader. Specifically, the reader organizes all IDs in a binary tree whose height is equal to the length of a tag ID. The left (right) branches of the tree is marked by '0s' ('1s'). Clearly, each leaf corresponds to a potential tag ID. The reader queries the tags by broadcasting a prefix starting from the root of the binary tree. The tags whose IDs match the queried prefix will respond their ID information. If two or more tags respond simultaneously, signal collision will occur, the reader then generates two new query prefixes by appending a '0' and a '1' to the previous query prefix. The tags will be queried by these two new prefixes successively. On the other hand, if exactly one (or none) tag responds its ID information, the reader will successfully receive the corresponding ID (or receive nothing). Then, the new nearby prefix will be queried in the next time. This process continues until all the tags have been identified [41].

Besides the exact identification, the problem of estimating the cardinality of tags has also attracted great attention from the research community. The first literature about tag estimation was proposed by Kodialam *et al.* in [14]. The proposed Unified Simple Estimator (USE) and Unified Probabilistic Estimator (UPE) perform estimation based on the number of empty slots or that of collision slots in a frame, respectively. Qian *et al.* [16] exploited the hashing with geometric distribution to estimate the cardinality of tags and thus proposed the Lottery Frame (LoF) scheme. Zheng *et al.* proposed Probabilistic Estimation Tree (PET) to provide a estimation method for the RFID systems which work based on tree-walking algorithms [18]. M. Shahzad *et al.* proposed the Average Run based Tag estimation (ART) by observing the average length of sequences of consecutive non-empty slots [19]. Li *et al.* proposed an estimation scheme called Maximum Likelihood Estimator (MLE) which takes the energy-efficiency into consideration [15]. These estimation schemes concentrate on approximating the cardinality of tags in a static RFID system. However, in practice, the RFID systems are usually dynamic—the tagged items or humans may frequently move in and out. The above estimation schemes can only tell you, for example, there are $10,000$ tags in the system at time T_1 and $15,000$ tags at time T_2. However, they cannot tell you how many tags are moved out and how many new ones are moved in during this period. Q. Xiao *et al.* studied the problem of tag estimation focusing on dynamic RFID systems [22]. ZDE scheme needs the reader to observe all slots in a time frame, which triggers its low time-efficiency. Gong *et al.* proposed INformative Counting (INC) to estimate the number of counterfeit tags whose IDs do not appear in the database [23].

VII. Conclusion

This paper has studied a practically important problem of key tag counting, which is very desirable in many application scenarios such as counting the key items in a store to facilitate

the restocking process, counting the rare birds to investigate their migration. To address this problem, we first proposed a *Basic Key tag Counting (B-KC)* protocol, whose good feature is that the reader only needs to observe the expected singleton slots instead of the whole time frame. To save time, B-KC skips the execution of expected empty/collision slots. Based on B-KC, we have exploited the sampling idea and early termination tactic to further proposed the *Sampling-based Key tag Counting (S-KC)* protocol, which possesses better efficiency. This paper has also theoretically investigated the parameter settings to guarantee the required estimation accuracy. Extensive simulation experiments have been conducted to evaluate the efficiency of the proposed S-KC. The results manifest that this scheme significantly outperforms the closely related protocols in terms of execution time. Moreover, the implemented prototype of S-KC demonstrates the deployability of our protocol.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61225010); NSFC under Grant nos. of 61173160, 61173161, 61173162, 61173165, 61272417, 61300187, 61300189, 61370198 and 61370199; Project funded by China Postdoctoral Science Foundation (Grant No. 2013M530916), the Fundamental Research Funds for the Central Universities (Grant No.DUT13ZD101 and DUT13JS04), New Century Excellent Talents in University (NCET-10-0095) of Ministry of Education of China, and HK RGC PolyU (5286/12E).

REFERENCES

[1] C.-H. Lee and C.-W. Chung, "Efficient Storage Scheme and Query Processing for Supply Chain Management Using RFID," *Proc. of ACM SIGMOD*, 2008.

[2] F. R. B. Nath and R. Want, "RFID Technology and Applications," *Proc. IEEE Pervasive Computing*, 2006.

[3] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: Indoor Location Sensing Using Active RFID," *Wireless networks*, vol. 10, no. 6, pp. 701–710, 2004.

[4] S. Preradovic, I. Balbin, N. C. Karmakar, and G. F. Swiegers, "Multiresonator-based Chipless RFID System for Low-cost Item Tracking," *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, no. 5, pp. 1411–1419, 2009.

[5] S. Lee, S. Joo, and C. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," *Proc. of IEEE MobiQuitous*, 2005.

[6] F. C. Schoute, "Dynamic Frame Length ALOHA," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 565 – 568, 1983.

[7] L. G. Roberts, "Aloha Packet System with and without Slots and Capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[8] J. Myung and W. Lee, "Adaptive Splitting Protocols for RFID Tag Collision Arbitration," *Proc. of ACM MobiHoc*, 2006.

[9] N. Bhandari, A. Sahoo, and S. Iyer, "Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency," *Proc. of IEEE ICIT*, 2006.

[10] V. Namboodiri and L. Gao, "Energy-Aware Tag Anti-Collision Protocols for RFID Systems," *Proc. of IEEE PerCom*, 2007.

[11] Y. Zheng and M. Li, "Fast Tag Searching Protocol for Large-Scale RFID Systems," *Proc. of IEEE ICNP*, 2011.

[12] M. Chen, W. Luo, Z. Mo, S. Chen, and Y. Fang, "An Efficient Tag Search Protocol in Large-Scale RFID Systems," *Proc. of IEEE INFOCOM*, 2013.

[13] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous Tracking using RFID tags," *Proc. of IEEE INFOCOM*, 2007.

[14] M. Kodialam and T. Nandagopal, "Fast and Reliable Estimation Schemes in RFID Systems," *Proc. of ACM MobiCom*, 2006.

[15] T. Li, S. Wu, S. Chen, and M. Yang, "Energy Efficient Algorithms for the RFID Estimation Problem," *Proc. of IEEE INFOCOM*, 2010.

[16] Y. L. C. Qian, H. Ngan and L. Ni., "Cardinality Estimation for Large-scale RFID Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1441–1454, 2011.

[17] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID Tags Efficiently and Anonymously," *Proc. of IEEE INFOCOM*, 2010.

[18] Y. Zheng and M. Li, "PET: Probabilistic Estimating Tree for Large-scale RFID Estimation," *IEEE Transactions on Mobile Computing*, vol. 11, no. 11, pp. 1763–1774, 2012.

[19] M. Shahzad and A. X. Liu, "Every Bit Counts - Fast and Scalable RFID Estimation," *Proc. of ACM MobiCom*, 2012.

[20] Y. Zheng and M. Li, "ZOE: Fast Cardinality Estimation for Large-scale RFID Systems," *Proc. of IEEE INFOCOM*, 2013.

[21] B. Chen, Z. Zhou, and H. Yu, "Understanding RFID Counting Protocols," *Proc. of ACM MobiCom*, 2013.

[22] Q. Xiao, B. Xiao, and S. Chen, "Differential Estimation in Dynamic RFID Systems," *Proc. of IEEE INFOCOM*, 2013.

[23] W. Gong, K. Liu, X. Miao, Q. Ma, Z. Yang, and Y. Liu, "Informative Counting: Fine-grained Batch Authentication for Large-Scale RFID Systems," *Proc. of ACM MobiHoc*, 2013.

[24] D. E. Smith, *A Source Book in Mathematics*. Courier Dover Publications, 2012.

[25] J. Rice, *Mathematical Statistics and Data Analysis*. Cengage Learning, 2006.

[26] N. V. Smirnov, I. V. Dunin-Barkovskij, and W. Richter, *Mathematische Statistik in der Technik*. Dt. Verlag d. Wiss., 1963.

[27] C. Qian, H. Ngan, and Y. Liu, "Cardinality Estimation for Large-scale RFID Systems," *Proc. of IEEE PerCom*, 2008.

[28] "EPC Radio-Frequency Identity Protocols Class-1 Gen-2 UHF RFID Protocol for Communications at 860MHz-960MHz, EPCglobal," *http://www.epcglobalinc.org/uhfclg2*, April 2011.

[29] T. Li, S. Chen, and Y. Ling, "Identifying the Missing Tags in a Large RFID System," *Proc. of ACM MobiHoc*, 2010.

[30] P. Semiconductors, "I-CODE Smart Label RFID Tags," *http://www.nxp.com/acrobat_download/other/identification/SL092030.pdf*, Jan 2004.

[31] E. EPCglobal, "Radio-frequency identity protocols class-1 generation-2 uhf rfid protocol for communications at 860 mhz–960 mhz version 1.0. 9," *K. Chiew et al./On False Authenticationsfor C1G2 Passive RFID Tags*, vol. 65, 2004.

[32] M. Shahzad and A. X. Liu, "Probabilistic Optimal Tree Hopping for RFID Identification," *Proc. of ACM SIGMETRICS*, 2013.

[33] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient Polling Protocols in RFID Systems," *Proc. of ACM MobiHoc*, 2011.

[34] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, "A Time-efficient Information Collection Protocols for Large-scale RFID Systems," *Proc. of IEEE INFOCOM*, 2012.

[35] W. Luo, S. Chen, T. Li, and S. Chen, "Efficient Missing Tag Detection in RFID Systems," *Proc. of IEEE INFOCOM*, 2011.

[36] W. Luo, S. Chen, T. Li, and Y. Qiao, "Probabilistic Missing-tag Detection and Energy-Time Tradeoff in Large-scale RFID Systems," *Proc. of ACM MobiHoc*, 2012.

[37] L. Yang, J. Han, C. Wang, T. Gu, and Y. Liu, "Season: Shelving Interference and Joint Identification in Large-scale RFID Systems," *Proc. of IEEE INFOCOM*, 2011.

[38] S. Tang, J. Yuan, M. Li, G. Chen, Y. Liu, and J. Zhao, "Raspberry: A Stable Reader Activation Scheduling Protocol in Multi-reader RFID Systems," 2009, pp. 304–313.

[39] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An Anticollision Algorithm for the Reader Collision Problem," 2003, pp. 1206–1210.

[40] "http://www.nordicsemi.com/eng/products/2.4ghz-rf/nrf24le1."

[41] M. Lehtonen, F. Michahelles, and E. Fleisch, "How to Detect Cloned Tags in a Reliable Way from Incomplete RFID Traces," *Proc. of IEEE RFID*, 2009.