

# A Parallel Identification Protocol for RFID Systems

Linghe Kong<sup>\*†</sup>, Liang He<sup>\*</sup>, Yu Gu<sup>\*</sup>, Min-You Wu<sup>†</sup>, Tian He<sup>‡</sup>

<sup>\*</sup>Singapore University of Technology and Design, Singapore

<sup>†</sup>Shanghai Jiao Tong University, China

<sup>‡</sup>University of Minnesota, USA

<sup>\*</sup>{linghe\_kong, he\_liang, jasongu}@sutd.edu.sg, <sup>†</sup>{linghe.kong, mwu}@sjtu.edu.cn, <sup>‡</sup>tianhe@cs.umn.edu

**Abstract**—Nowadays, RFID systems have been widely deployed for applications such as supply chain management and inventory control. One of their most essential operations is to swiftly identify individual tags to distinguish their associated objects. Most existing solutions identify tags sequentially in the temporal dimension to avoid signal collisions, whose performance degrades significantly as the system scale increases. In this paper, we propose a *Parallel Identification Protocol* (PIP) for RFID systems, which achieves the parallel identification paradigm and is compatible with current RFID devices. Uniquely, PIP encodes the tag ID into a specially designed pattern and thus greatly facilitates the reader to correctly and effectively recover them from collisions. Furthermore, we analytically investigate its performance and provide guidance on determining its optimal settings. Extensive simulations show that PIP reduces the identification delay by about 25% – 50% when compared with the standard method in EPC C1G2 and the state-of-the-art solutions.

## I. INTRODUCTION

Radio Frequency IDentification (RFID) systems are rapidly changing our lives with their low costs and ubiquitous characteristics. They have been widely deployed in various applications such as inventory control [4], supply chain management [8], and object tracking [21]. In order to identify the associated object for each tag, one essential operation in RFID systems is for the reader to acquire the unique IDs of individual tags through the communication between them, known as the *tag identification* [24]. The performance of the identification process is normally evaluated by the time for it to be accomplished, referred to as the *identification delay*.

In RFID systems, the communication starts with the reader transmitting a high power waveform query, and the tags modulate the received signal by changing the impedance match on its own antenna and reflect back to the reader [3]. Due to this special property of RFID communication, it is not feasible for tags to detect whether other tags are communicating with the reader, *e.g.*, through channel sensing as in traditional wireless communications [4]. Thus to avoid the collisions caused by the simultaneous transmission of multiple tags, tag identification is traditionally accomplished by assigning individual tags with different time slots to inform the reader of their IDs, and the identification process is carried out in a sequential manner [13], [15]. These existing sequential identification solutions demonstrate great efficiency for RFID systems with small and medium scales. However, their performance degrades dramatically in large-scale RFID systems because the tags need to be identified one-by-one.

To improve the identification performance in large-scale

RFID systems, two kinds of parallel solutions have been designed: the conventional CDMA-based identification solutions [12], [19] and the Buzz code proposed in [18]. The CDMA-based RFID systems are known to be expensive and power hungry [6], while Buzz imposes additional requirement on system to accurately differentiate the signal strength, which is beyond the hardware requirement of EPC Class-1 Generation-2 (C1G2) [3] standard or current off-the-shelf RFID readers.

To efficiently achieve the parallel identification while guaranteeing its compatibility with current off-the-shelf RFID devices, we propose a novel *parallel identification protocol* in this paper. The motivation of our design is simple: although the collisions from multiple tags are normally considered undesirable for communications, partial original information is conserved in the collision. Our main contributions of this paper are four-fold.

- We design a novel *L-K code*, which encodes the tag ID into a bit string with a specially designed pattern. This pattern facilitates the reader in recovering part of the ID information from collided signals.
- As only partial information can be recovered if tag ID is encoded by the *L-K code* and responded to the reader, we further propose a *randomized mapping scheme*, with which the *L-K code* can be applied to the same tag multiple times with different inputs. As a result, multiple partially recovered ID information can be jointly utilized to recover the full information.
- Combining the *L-K code* and the randomized mapping scheme, we propose a *Parallel Identification Protocol* (PIP) for RFID systems to identify tags in a parallel manner, which significantly reduces the identification delay especially for large-scale RFID systems.
- We analytically investigate the performance of PIP and provide guidance on identifying its optimal settings. Our evaluation results show that PIP can achieve about 25% – 50% reduction of the identification delay when compared with classic identification solutions.

The rest of this paper is organized as follows. The related work on tag identification is presented in Section II. We formulate the problem in Section III. The detailed design of PIP is presented in Section IV, and we investigate the optimal protocol setting in Section V. A few practical issues are discussed in Section VI. We present the evaluation results in Section VII, and conclude this paper in Section VIII.

## II. RELATED WORK

Broadly, existing identification methods can be classified into two categories: *sequential* and *parallel* identification.

### A. Sequential Identification Methods

Most current identification works focus on the sequential methods, which carry out in a TDMA manner [9]: the tags' responses are scheduled at different time slots by the reader with polling [15], splitting tree [13], frame slotted ALOHA (FSA) [20], [26], etc. In current C1G2 [3],  $Q$ -protocol is adopted to be the standard identification method, which is a dynamic FSA protocol and  $Q$  is a parameter to adjust the length of the next frame. If collision occurs, the affected tags will restart during the next frame. To the best of our knowledge, the state-of-the-art sequential method so far is BIC [23], in which the tags hash and transmit their IDs to multiple frames, and the identification is accomplished based on a constructed Bloom Filter [10].

The advantage of the sequential methods is in reducing the transmission collisions from multiple interrogated tags. However, these methods introduce additional scheduling overhead, redundant frame slots, and guard interval, and thus prolong the identification process especially for large-scale RFID systems.

### B. Parallel Identification Methods

Just a few identification works consider the parallel methods, which potentially carry out the identification process based on SDMA [4], FDMA [6], or CDMA [12], [18], [19].

Our PIP is orthogonal to SDMA [4] and FDMA [6], thus they can work together to reduce the identification delay.

The proposed PIP is one coding based parallel identification protocol but totally different from existing CDMA methods. A conventional CDMA-based identification method is proposed in [12] for RFID systems, which requires tags to multiply their IDs with a pseudorandom noise before transmission. As a result, the transmitted ID is prolonged, and additional identification delay is introduced. Our evaluation results show that PIP outperforms the CDMA-based method [12] by 45% in terms of reducing the identification delay.

Recently, Wang et al. designed an advanced Buzz code [18] based identification system. Buzz code formats the tags' responses into customized patterns and resolves the collisions with compressive sensing. However, with Buzz, the reader needs to accurately differentiate the signal strength, which is beyond the hardware requirement of C1G2 [3] standard or current off-the-shelf RFID readers. Unlike Buzz [18], PIP achieves the parallel identification of RFID tags without requiring any hardware modifications.

## III. PRELIMINARY

### A. System Model

We consider a RFID system with a single reader and a number of tags. The reader has relatively powerful computation and storage capability, and it can access the database that stores all tag IDs in the system [22]. On the other hand, the computation capabilities of tags are quite limited [23].

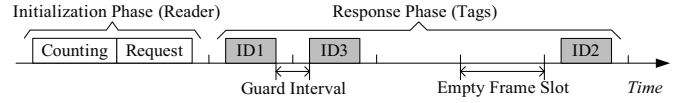


Fig. 1. Overview of typical sequential identification methods.

In RFID systems, the communication between the reader and tags starts with the reader transmitting a high power waveform query. Then, the tags respond by reflecting back the waveform using the ON-OFF keying: tags respond a bit '1' by reflecting the waveform and a bit '0' by keeping silent [4]. The tags in the communication range of a reader are normally referred as *interrogated tags*. Denote  $m$  and  $n$  as the total number of tags in the system and the number of interrogated tags to be identified, respectively.

Every RFID tag has a predefined binary string to store their basic information, with typical lengths from 64 to 512 bits [3]. In a given system, partial bits in the binary string construct the unique tag ID, which is adequate to distinguish  $m$  tags in the system, i.e.,  $B = \lceil \log_2 m \rceil$ . This  $B$ -bit binary string is easily converted into a decimal ID number, which is adopted in this paper to improve the readability. The identification operation in a RFID system therefore is to obtain the IDs of all interrogated tags at the reader side [4].

### B. Motivations and Problem Statement

In most existing tag identification solutions [20], [23], [26], to avoid the potential collisions from multiple interrogated tags, the reader schedules the tags' responses in different frame slots, and obtain their IDs in a sequential one-by-one manner. Note that for presentation clarity, we refer to *frame slot* as the time duration to transmit one entire tag ID, and *bit slot* is referred to as the time duration to transmit a single bit. Figure 1 presents a high-level overview of the sequential identification methods, in which the reader first initializes the identification process (e.g., estimates the number of interrogated tags and schedules their responses) and then the tags respond at different frame slots.

However, this sequential identification paradigm leads to a long identification delay due to following two reasons: (i) to reduce the transmission collisions, the total number of frame slots is normally much larger than the number of interrogated tags (e.g., in the FSA based protocols [20], [23], [26]), which leads to many wasted frame slots. (ii) Furthermore, *guard intervals* [3] are adopted to ensure the consecutive responses from the two tags are separated from each another.

Observing the limitations of the existing sequential identification methods, in this paper we aim to design a parallel identification method that is able to obtain the IDs of interrogated tags within the shortest possible delay, which is defined as the time since the reader starts the identifying process till all the IDs have been obtained. In RFID systems, the length of a bit slot is relatively stable, e.g., about  $16\mu s$  in the C1G2 standard [3]. Thus for simplicity, we adopt the number of bit slots to describe the identification delay in this paper unless otherwise specified.

TABLE I  
LIST OF THE MAPPING AND ENCODING RESULTS OF  $m = 10$  TAGS  
BY RANDOMIZED MAPPING SCHEME AND  $L$ - $K$  CODE  
WHEN SETTING  $L = 5$ ,  $K = 2$ , NUM OF SEGMENTS = 2

1-st segment		2-nd segment	
ID mapping	ID encoding	ID mapping	ID encoding
1 $\rightarrow$ 1	$\rightarrow$ 11000	1 $\rightarrow$ 5	$\rightarrow$ 01100
2 $\rightarrow$ 2	$\rightarrow$ 10100	2 $\rightarrow$ 8	$\rightarrow$ 00110
3 $\rightarrow$ 3	$\rightarrow$ 10010	3 $\rightarrow$ 1	$\rightarrow$ 11000
4 $\rightarrow$ 4	$\rightarrow$ 10001	4 $\rightarrow$ 4	$\rightarrow$ 10001
5 $\rightarrow$ 5	$\rightarrow$ 01100	5 $\rightarrow$ 7	$\rightarrow$ 01001
6 $\rightarrow$ 6	$\rightarrow$ 01010	6 $\rightarrow$ 10	$\rightarrow$ 00011
7 $\rightarrow$ 7	$\rightarrow$ 01001	7 $\rightarrow$ 3	$\rightarrow$ 10010
8 $\rightarrow$ 8	$\rightarrow$ 00110	8 $\rightarrow$ 6	$\rightarrow$ 01010
9 $\rightarrow$ 9	$\rightarrow$ 00101	9 $\rightarrow$ 9	$\rightarrow$ 00101
10 $\rightarrow$ 10	$\rightarrow$ 00011	10 $\rightarrow$ 2	$\rightarrow$ 10100

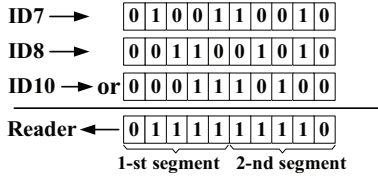


Fig. 2. The received bits are collided by concurrent responses of three tags.

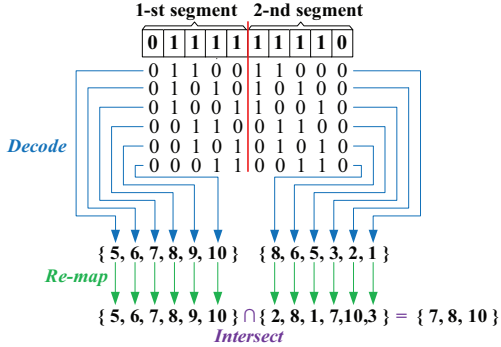


Fig. 3. Three tags {7,8,10} are resolved from the received bits.

#### IV. PARALLEL IDENTIFICATION PROTOCOL

In this section, we first use a simple case study to present a high level overview of the proposed *parallel identification protocol* (PIP), and then we explain its operations in detail.

##### A. Case Study

We utilize a small-scale case study to show the essential idea of parallel identification. Assume there are  $m = 10$  tags in a system with IDs  $\{1, 2, \dots, 10\}$ , and three interrogated tags  $\{7, 8, 10\}$  need to be identified.

At the beginning of the identification process, the reader initializes the setting by “encoding every ID into a 5-bit string with 2 bits being ‘1’s”, and broadcasts an identification request with the setting. (Here we assume Table I has been obtained and satisfies the setting requirement. In Section IV-D, the computation of this table by tags will be introduced in detail.) After receiving the request from the reader, tag  $\{7\}$  maps its ID  $7 \rightarrow 7$  and encodes it into 01001 as its 1-st segment, as shown in Table I. For the 2-nd segment, tag  $\{7\}$  maps  $7 \rightarrow 3$  and encodes it into 10010. As a result, the signals replied by tag  $\{7\}$  is 0100110010. Similarly, tags  $\{8, 10\}$  reply with 0011001010

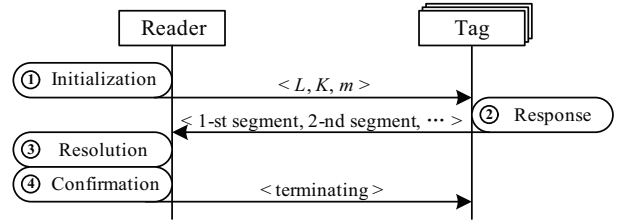


Fig. 4. Flow chart of parallel identification protocol.

and 0001110100 respectively. Due to the simultaneous transmission, the signals received by the reader is 011111110, as shown in Fig. 2. The reader resolves the IDs as shown in Fig.3. Based on the 1-st received segment 01111 and the 2 ‘1’s setting, the reader finds that there are six possible transmitted bit strings:  $\{01100, 01010, 01001, 00110, 00101, 00011\}$ . Decoding these bit strings and re-mapping the IDs according to Table I, the reader obtains a candidate interrogated tag set of  $\{5, 6, 7, 8, 9, 10\}$ . Similarly, a candidate interrogated tag set of  $\{2, 8, 1, 7, 10, 3\}$  can be obtained based on the 2-nd received segment. Then the reader obtains the IDs of the three interrogated tags by intersecting the two candidate sets:  $\{5, 6, 7, 8, 9, 10\} \cap \{2, 8, 1, 7, 10, 3\} = \{7, 8, 10\}$ . At last, the reader checks whether tag  $\{7, 8, 10\}$  are the actual interrogated tags, and terminates the identification if they are confirmed.

##### B. Protocol Overview

From the above case study, we can see the identification process with PIP consists of four phases as shown in Fig. 4: *initialization*, *response*, *resolution*, and *confirmation*. During the initialization phase, the reader initializes the protocol settings and triggers the identification process by broadcasting a request. During the response phase, the interrogated tags encode their IDs to a bit string with the proposed *randomized mapping scheme* and *L-K code* (which are two key components of PIP), and then they concurrently transmit the encoded bit strings to the reader until a *terminating* command is received. The resolution phase is performed by the reader, and it overlaps with the response phase in the temporal dimension. While continuously receiving the transmitted bit string from tags, the reader resolves the received signals by dividing them into segments. And at the end, a *terminating* command is broadcasted to stop the identification process when all the ID information are confirmed in the confirmation phase.

In the remaining of this section, we elaborate each of the four phases in detail.

##### C. Phase 1: Initialization

The objective of the initialization phase is to determine the optimal settings and to trigger the identification process. There are three parameters  $\langle L, K, m \rangle$  needed to be transmitted to tags, which are used to encode the IDs during the response phase, where  $L$  is the length of segment,  $K$  is the number of ‘1’s in every segment, and  $m$  is the total number of tags. *e.g.*,  $L = 5$ ,  $K = 2$ , and  $m = 10$  in the above case study.

The parameter  $m$  is available to the reader, and  $\langle L, K \rangle$  are jointly determined by  $m$  and the number of interrogated

tags  $n$ . There are many efficient methods to estimate the tag number  $n$  in the literatures [7], [14], [16], [25]. For example, the advanced estimation method ART [16] can estimate 5,000 tags within 0.5 second with an error ratio  $\leq 5\%$  and a reliability  $\geq 90\%$ . We emphasize that PIP does not require a perfect estimation of  $n$ . In the confirmation phase, we have taken the possible estimation error into the design consideration. Furthermore, our evaluation results indicate that PIP can tolerate an estimation error of  $n$  up to 10%, which can be easily achieved by existing estimation solutions [16], [25].

For presentation clarity, we delegate the detailed derivation on optimal  $\langle L, K \rangle$  by  $m$  and  $n$  to Section V, and assume the parameters has been optimally initialized here.

After determining the parameters, the reader broadcasts  $\langle L, K, m \rangle$  to the interrogated tags and triggers the *response* phase at the tag side.

#### D. Phase 2: Response

After receiving the request, the interrogated tags encode their IDs and transmit them as the response. To encode their IDs, the tags first map their IDs with parameter  $m$  using a randomized mapping scheme and then encode the mapped IDs with parameter  $L$  and  $K$  using  $L$ - $K$  code. This *mapping-and-encoding* process is repeated to obtain the bit string catenated by segments. The tags keep transmitting the encoded bit strings to the reader until a *terminating* message is received.

1) *Randomized Mapping Scheme*: The objective of our mapping scheme is for tags to generate different ID numbers in different segments.

The basic requirement of a mapping scheme in PIP is bijective, *i.e.*, injective and surjective [2]. The injection indicates that every ID has a unique mapped ID (one-to-one), so that the mapped ID can re-map to the original ID during the resolution phase. The surjection guarantees that the original space and the mapped space have the same number of  $m$  IDs, thus no redundant ID number will be introduced.

Two intuitive bijective mapping methods, referred to as *shift mapping scheme* and *enhanced shift mapping*, are shown in Table II when  $m = 10$ . The shift mapping scheme utilizes +1 rotation method. Compared with shift mapping scheme, the enhanced shift mapping scheme has the same results in the odd segments and the reverse permutation in the even segments. However, such mapping schemes have the obvious equal-difference pattern, which leads to several non-interrogated IDs appearing with high probability in candidate sets, and thus more segments are needed to filter them out. *e.g.*, it costs 5 and 4 segments respectively that using PIP with shift mapping and enhanced shift mapping to resolve  $\{7, 8, 10\}$  in the case study. Thus, we propose to utilize the unordered random mapping for PIP to reduce the impact from any ordered pattern.

Inspired by the current linear congruential pseudo random number generator [11], we design a *randomized mapping scheme*. Denote the ID of a tag as  $I$  ( $I = 1, 2, \dots, m$ ), and use  $I(i)$  to represent the mapping result of  $I$  for the  $i$ -th segment. With randomized mapping scheme, every tag  $I$  distributively

TABLE II  
LIST THE RESULTS OF DIFFERENT MAPPING SCHEMES WHEN  $m = 10$

Shift			Enhanced Shift			Randomized		
1-st	2-nd	3-rd	1-st	2-nd	3-rd	1-st	2-nd	3-rd
1	2	3	1	1	3	1	5	10
2	3	4	2	10	4	2	8	7
3	4	5	3	9	5	3	1	4
4	5	6	4	8	6	4	4	1
5	6	7	5	7	7	5	7	8
6	7	8	6	6	8	6	10	5
7	8	9	7	5	9	7	3	2
8	9	10	8	4	10	8	6	9
9	10	1	9	3	1	9	9	6
10	1	2	10	2	2	10	2	3

computes its mapped ID  $I(i)$  by

$$I(i) = \begin{cases} I & i = 1, \\ (I \times S_{i-1} + i - 1) \% m + 1 & i \geq 2, \end{cases} \quad (1)$$

where  $\%$  is the modulo operation and  $S_{i-1}$  is the mapping seed for computing  $I(i)$ . The set of mapping seed  $S$  is a sequence of sorted integers, in which every seed is relatively prime to  $m$ . For example,  $S = [3, 7, 9, 11, \dots]$  when  $m = 10$ . In the case study, tag  $\{7\}$  maps its ID  $I = 7$  to its mapped ID  $I(1) = 7$  for the 1-st segment, *i.e.*,  $I(1) = I$  when  $i = 1$  according to Eq. (1). And tag  $\{7\}$  maps  $I = 7$  to  $I(2) = 3$  for the 2-nd segment, *i.e.*,  $i = 2$  and  $S_{2-1} = S_1 = 3$ , then  $I(2) = (7 \times 3 + 2 - 1) \% 10 + 1 = 3$  according to Eq. (1). All  $m = 10$  IDs' mapping results for the first 3 segments are shown in Table II, which shows pseudo random permutation.

Based on Definition 2.1.11 (Complete Set of Residues) and Lemma 2.1.12 in [17], it can be proved that the randomized mapping scheme is bijective.

The computational complexity of Eq. (1) is  $\mathcal{O}(1)$  and computing one co-prime seed to  $m$  is  $\mathcal{O}(\log m)$ . Hence, the complexity of the randomized mapping scheme is  $\mathcal{O}(\log m)$ .

2) *L-K Code*: The objective of coding scheme is to convert IDs into bit strings for transmission. We propose a novel  $L$ - $K$  Code scheme with which a tag encodes its ID into a  $L$ -bit segment with  $K$  '1's and  $(L - K)$  '0's according to the given  $L$  and  $K$ . This specially designed coding pattern facilitates the reader to recover the partially conserved ID information from the collided signals.

Denote  $b_j$  as the  $j$ -th bit in a segment. We introduce  $k$  to record the remainder number of '1's, which is initialized by  $k = K$ . We also introduce  $z_j$  as a referential number to  $b_j$ , where  $z_0 = 0$  is just for initialization.  $L$ - $K$  encoding operates from the first (left hand side) bit  $b_1$  to the last bit  $b_L$ . For every  $b_j$ , firstly, the referential number is updated by  $z_j = z_{j-1} + \binom{L-j}{k-1}$ . Secondly, the value of ID and the referential number is compared. If  $I > z_j$ , then  $b_j$  is set to '0'. If  $I < z_j$ , then  $b_j = '1'$ , the  $z_j$ 's update for this bit is cancelled  $z_j = z_{j-1}$ , and  $k$  is updated by  $k = k - 1$ . If  $I = z_j$ , then  $b_j$  and the last  $k - 1$  bits in this segment are set to '1', and the bits from  $b_{j+1}$  to  $b_{j-k+1}$  are all set to '0'.

For example, in the 1-st segment of the case study, the encoding operation of tag  $\{7\}$  is as follows: known  $L = 5, k = K = 2, I = 7$ . For  $b_1, z_1 = 0 + \binom{5-1}{2-1} = 4$ , we find  $I > z_1$ , so  $b_1 = '0'$ . For  $b_2, z_2 = 4 + \binom{5-2}{2-1} = 7$ , we find  $I = z_2$ , so



$b_2=1$  and the last  $2 - 1 = 1$  bit *i.e.*,  $b_5=1$ , the values from  $b_3$  to  $b_4$  are all '0's. Hence, tag {7} is encoded to 01001.

For each  $b_j$ , the complexity to compute  $\binom{L-j}{k-1}$  is  $\mathcal{O}(\log L)$ . Since the algorithm runs from  $b_1$  to  $b_L$ , the computational complexity of the  $L$ - $K$  encoding algorithm is  $\mathcal{O}(L \log L)$ .

#### E. Phase 3: Resolution

In the *resolution* phase, the reader manages to resolve the IDs upon the reception of every  $L$  bits (*i.e.*, every segment). However, the directly resolved IDs may have some errors due to the collision of simultaneous responses. Resolution in PIP jointly utilizes the resolved information from all previously received segments to remove the resolving errors.

A reader resolves the IDs from the received collision in 4 steps. First, for every segment, it needs to *list* all possible combinations. Second, *decode* each combination to obtain the mapped IDs, then *re-maps* them to be a set of candidate IDs, and finally *intersect* candidate ID sets of different segments to converge to the interrogated IDs.

*List step:* With the proposed  $L$ - $K$  code, we know that if there are  $K' \geq K$  '1's in a received segment, the number of  $L - K$  combinations is

$$\binom{K'}{K}. \quad (2)$$

In our case study shown in Fig. 3, both the 1-st segment 01111 and the 2-nd segment 11110 have  $K' = 4$  '1's. Known  $K = 2$ , there are  $\binom{4}{2} = 6$  combinations in both segments respectively, the bit strings of the combinations are listed in Fig. 3.

*Decode step:* The decoding operation is to convert all combinations in one segment to a set of mapped IDs. Different from the tags, which requires to encode online, a reader typically has sufficient storage capacity. The decoding operation can be easily accomplished by pre-storing the offline computed  $L$ - $K$  code table in the reader's memory and check the table for the corresponding ID when decoding. Table I shows the code table for our case study. By looking up the table, the combinations of two segments can be decoded to the sets of mapped IDs  $\{5,6,7,8,9,10\}$  and  $\{8,6,5,3,2,1\}$  respectively.

Note that it is also feasible for the reader to perform online decoding with low computation in very large RFID systems (where  $m$  may be too large for the corresponding code table to be stored in a reader's memory). However, we do not include the detailed description on the online decoding algorithm here due to the space constraints.

*Re-map step:* The re-map step is the inverse operation of mapping. After obtaining a set of mapped IDs, the reader needs to re-map them to the original IDs of associated tags, which again can be accomplished by storing a ID table in the reader's memory, as shown in Table II. In our case study shown in Fig. 3, two sets of mapped IDs can be re-mapped to their original IDs and form the candidate sets, *i.e.*,  $\{5,6,7,8,9,10\}$  and  $\{8,6,5,3,2,1\}$  are re-mapped to  $\{5,6,7,8,9,10\}$  and  $\{2,8,1,7,10,3\}$  by the re-mapping operation.

*Intersection step:* False IDs (*i.e.*, the IDs do not belong to any interrogated tags) may exist in the candidate sets due to

1-st segment						1-st segment						2-nd segment					
0 1 1 1 1						0 1 1 1 1						1 1 1 1 0					
5	0	1	1	0	0	7	0	1	0	0	1	1	0	0	1	0	0
6	0	1	0	1	0	8	0	0	1	1	0	0	1	0	1	0	1
7	0	1	0	0	1	10	0	0	0	1	1	1	0	0	1	0	0
8	0	0	1	1	0		0	1	1	2	2	2	1	0	0	0	0
9	0	0	1	0	1								2	1	1	2	0
10	0	0	0	1	1												
0 3 3 3 3																	

Fig. 5. Three tags {7,8,10} are confirmed.

the noise caused by transmission collisions. In our design, the reader eliminates the false IDs by jointly utilizing the candidate sets obtained from multiple segments. This is based on the observation that *the true IDs exist in every candidate set, while the false IDs obtained from different segments are random due to the randomized mapping scheme*. As a result, the reader is able to filter out the false IDs by *intersecting* the candidate set from each segment. In the case study in Fig. 3, where the true IDs are  $\{7, 8, 10\}$ , the false IDs obtained from the 1-st and 2-nd segments are  $\{5, 6, 9\}$  and  $\{2, 1, 3\}$  respectively, then the reader can obtain the true IDs by intersecting the two sets:  $\{5, 6, 7, 8, 9, 10\} \cap \{8, 6, 5, 3, 2, 1\} = \{7, 8, 10\}$ .

#### F. Phase 4: Confirmation

The last question to be addressed is how the reader confirms the ID information is correctly resolved and then terminates the identification process. An intuitive method is to terminate the identification process when there are  $\tilde{n}$  IDs in the intersected results, where  $\tilde{n}$  is the estimation number of interrogated IDs by the cardinality estimation in the initialization phase. However, this is not a reliable approach because the estimated  $\tilde{n}$  may deviate from the ground truth  $n$ .

We design a confirmation operation based on analyzing the received segments, which requires only a little computation on the reader but does not require any additional communication between the reader and the tags. The confirmation operation takes advantage of the observation that: Every '1' in the received segment indicates at least one ID needs to transmit '1' at that bit slot. Once it ensures that only one certain ID transmits '1' at that slot, then that '1' can serve as an evidence to confirm the ID. When all resolved IDs can be confirmed, the identification process terminates.

Still take the case study in Section IV-A as an example. After the 1-st segment, the confirmation operation converts the resolved IDs  $\{5,6,7,8,9,10\}$  to bit strings according to the *mapping-and-encoding* process using in the 1-st segment, and then forms the bit strings to be a matrix as shown in Fig. 5. Sum the elements in each column, we obtain the summed vector 03333. Since '0' means no ID transmission and '3' means 3 possible tags' transmission at that slot, the reader cannot confirm any unique ID based on the 1-st segment only. Hence, the confirmation need to repeat after the 2-nd segment, where the resolved IDs become  $\{7,8,10\}$  after intersection. The confirmation converts them and forms two matrices corresponding to 2 segments respectively as shown in Fig. 5. This time, the summed vector of the 1-st segment is 01122. The '1' at the 2-nd bit is used to confirm the

TABLE III  
LIST OF  $L$ - $K$  CODE WHEN  
 $m=10$   $L=5$   $K=2$

ID	$L-K$ code
1	11000
2	10100
3	10010
4	10001
5	01100
6	01010
7	01001
8	00110
9	00101
10	00011

TABLE IV  
OPTIMAL  $L$  AND  $K$  FOR DIFFERENT  $n$   
WHEN  $m=10000$

$n$	$L$	$K$	$r$	$T=L \times r$
1	16	7	1	16
2	24	4	2	48
5	24	4	5	120
10	61	3	4	244
20	96	3	5	480
50	151	2	8	1208
100	264	2	9	2376
200	607	2	8	4856
500	10000	1	1	10000
10000	10000	1	1	10000

ID {7}. The reason is that from the candidate set, only ID {7} can transmit '1' at the 2-nd slot and the other IDs {8,10} transmit '0's at that slot. Hence, ID {7} is confirmed to be an interrogated tag. Similarly, the 3-rd bit in 01122 is utilized to confirm ID {8}. Then, the summed vector of the 2-nd segment is 21120. The ID {10} can be confirmed by the '1' at the 3-rd bit of the vector. Since all IDs in the candidate set are confirmed, the reader broadcasts the *terminating* command to stop the identification process.

Overall, the resolution phase is to reduce the number of candidate IDs by filtering the false IDs using intersection operation, and the confirmation phase is to confirm the true IDs and to provide the *terminating* command. These two phases together help the reader to identify IDs in the collision.

To reduce the computation overhead, the reader does not need to perform the confirmation operation every time a new segment is received. With the estimated  $\tilde{n}$ , the reader only tries to confirm the resolved IDs when their number is smaller than  $\eta\tilde{n}$ , where  $\eta$  is a given redundancy coefficient.

## V. DETERMINING THE OPTIMAL $\langle L, K \rangle$

In this section, we theoretically investigate the optimal parameters for  $L$ - $K$  code setting, and then derive the expectation of identification delay for PIP.

In PIP, the parameters  $\langle L, K \rangle$  are computed by the reader in the initialization phase. After receiving  $\langle L, K \rangle$  from the reader, the interrogated tags then can encode their IDs with the  $L$ - $K$  code and respond the encoded bit strings to the reader for identification purpose. These parameters directly affect how many bits need to be transmitted before the identification process can be terminated, *i.e.*, the identification delay. In the following we discuss how to determine their optimal settings.

Our design objective is to accomplish the identification process with the shortest possible time

$$\min(L \times r), \quad (3)$$

where  $r$  is the expected number of segments that is adequate to accomplish the identification process.

We will show that the feasible range of  $L$  is limited, and then both  $K$  and  $r$  can be determined with a given  $L$ . As a result, the optimal  $L$  with regard to Eq. (3) can be obtained.

When applying the  $L$ - $K$  code to a RFID system with  $m$  tags, it is clear that an  $L$  larger than  $m$  is not necessary, and

thus  $L \leq m$ . Furthermore, to guarantee each tag has a unique ID, it is clear the following requirement needs to be satisfied:

$$\binom{L}{K} \geq m. \quad (4)$$

With a given  $x$ , the maximal value of  $\binom{x}{y}$  is achieved when  $y = \frac{x}{2}$ , and thus we know a necessary condition for Eq. (4) to hold is

$$\binom{L}{\lfloor L/2 \rfloor} \geq m. \quad (5)$$

Since  $\binom{L}{\lfloor L/2 \rfloor}$  monotonically increases with  $L$ , the lower bound of  $L$  is the minimal value of  $L$  that satisfies Eq. (5). Thus the range of feasible  $L$  therefore is:

$$L \in \left[ \min\{L \mid \binom{L}{\lfloor L/2 \rfloor} \geq m\}, m \right]. \quad (6)$$

Next let us consider the parameter  $K$ . Because  $K$  determines the number of '1's in the encoded bit string, a larger  $K$  leads to a higher chance for transmission collisions to happen. Thus with a given  $L$ ,  $K$  is desired to be as small as possible while guaranteeing Eq. (4), and thus the optimal  $K$  with a given  $L$  can be determined by

$$\min\{K \mid \binom{L}{K} \geq m\}. \quad (7)$$

To investigate how to determine  $r$  with a given  $L$ , let us first consider the expected number of '0's in each received segment at the reader, which is denoted as  $E_0$ . The  $L$ - $K$  code has an important property that links  $r$  with any given  $L$ .

**Property 1:** With  $L$ - $K$  code, the probability for each bit in an encoded segment to be '1' is  $q = \frac{K}{L}$ .

To present a clear observation on Property 1, Table III shows the encoded bit strings of 10 IDs with  $L=5$  and  $K=2$ . We can see that the 1-st bits of the strings for ID {1,2,3,4} are '1's, and the 1-st bits of ID {5,6,7,8,9,10} are '0's. Hence, the probability of '1' for this bit is  $\frac{4}{10} = \frac{2}{5} = \frac{K}{L}$ . Similar pattern can be observed at the 2-nd, 3-rd, 4-th, and 5-th bits.

For any bit of the received segment at the reader, it is '0' if and only if none of  $n$  interrogated tags response '1' at that bit slot. Thus based on Property 1, the probability  $p_0$  for any bit slot in the received segment is '0' is:

$$p_0 \approx \prod_{i=0}^{n-1} \frac{m(1-q) - i}{m - i}, \quad (8)$$

and then  $E_0$ , which is denoted the expected number of '0's in one received segment, can be computed as:

$$E_0 = L \times p_0. \quad (9)$$

Consequently, we can calculate the expected number of '1's in the received segment as  $E_1 = L - E_0$ , and thus according to Eq. (2), the expected number of IDs decoded from each received segment is

$$C = \binom{E'_1}{K}, \quad (10)$$

where  $E'_1$  is the rounded value of  $E_1$ .

Then, the number of false IDs can be calculated as  $(C - n)$ . Since  $(C - n)$  false IDs are randomly generated from  $(m - n)$  total possible false IDs, after intersecting two sets of  $(C - n)$  random IDs from two different segments, the expected number of remaining false IDs is  $(m - n) \left( \frac{C - n}{m - n} \right)^2$ , where  $\frac{C - n}{m - n}$  is the probability for a specific false ID to appear in the resolved result. Thus, the number of false IDs remaining after intersecting  $r$  resolved results can be calculated as

$$F(r) = (m - n) \left( \frac{C - n}{m - n} \right)^r. \quad (11)$$

We need  $F(r) \rightarrow 0$  for the identification process to be terminated. Thus the minimal number of segments needed is

$$\min\{r \mid F(r) < \epsilon\}, \quad (12)$$

where  $\epsilon$  is a small constant.

Given an arbitrary  $L$ , the minimal  $r$  can be computed by Eq. (7)-(12), and then the identification delay is  $L \times r$  for this  $L$ . Since the feasible range of  $L$  is known by Eq. (6), we can search the minimal  $r$  for every  $L$ . The  $L$  in its range that its corresponding  $L \times r$  satisfies Eq. (3) is confirmed as the optimal value of  $L$ , and then the corresponding  $K$  can be determined according to Eq. (7).

When the optimal  $\langle L, K \rangle$  are obtained, the expected  $r$  can be determined by Eq. (12). Then the expectation of identification delay  $T$  is easily to be calculated by  $L \times r$ .

To summarize, the only information required to determine the optimal  $\langle L, K \rangle$  is the total number of tags in the system  $m$  and the number of interrogated tags  $n$ . For a specific RFID system, the total number of tags  $m$  is available to the reader. Thus to further speed up the identification process, we can pre-calculate the optimal  $\langle L, K \rangle$  for every possible  $n$ , and store the results in the reader's memory. An example on the pre-calculated values including optimal  $\langle L, K \rangle$ , expected  $r$ , and expected  $T$  for  $m = 10,000$  is shown in Table IV.

## VI. PRACTICAL ISSUE

In this section, we discuss a few practical issues when implementing PIP.

### A. Tags Synchronization

The proposed PIP requires the responses from tags are synchronized. It is reported in [18] that the synchronization offset for commercial RFID tags is normally less than  $1\mu s$ . Considering the low bit rates in RFID systems, e.g.,  $64Kbps$  [3], the  $1\mu s$  offset is only about  $1\mu s / (1s/64Kbps) = 6.4\%$  of the duration of a bit slot, and thus the synchronization offset has only a negligible impact on PIP.

### B. Computational Overhead

Another issue needs to be considered is whether the low computation capability of RFID tags is enough to perform PIP. With PIP, every tag needs to locally perform the random mapping and  $L$ - $K$  encoding. The computational complexities of randomized mapping scheme and  $L$ - $K$  code are  $\mathcal{O}(\log m)$  and  $\mathcal{O}(L \log L)$ , respectively. On one hand, even in a large

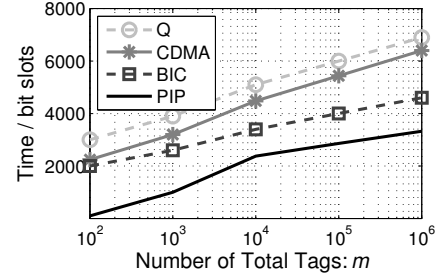


Fig. 6. Comparison on four protocols when  $m$  varying and  $n = 100$ .

scale RFID system with  $m = 10^6$  and  $n = 100$ , the optimal value of  $L$  is only 500 obtained by the equations in Section V, then  $\log m$  and  $L \log L$  are still small values in PIP. On the other hand, many current RFID tags such as Intel WISP [1] and EnHANTs [5] adopt the MSP430 process with a frequency of  $16 MHz$ , which is far sufficient to perform the required mapping and encoding operations in PIP.

## VII. PERFORMANCE EVALUATION

We evaluate the performance of PIP and provide more insights on its efficiency in this section.

### A. Simulation Setting

We simulate a  $m$ -tag RFID system with  $n$  interrogated tags, and a single reader is adopted to perform the identification task. Each tag has a unique integer ID in the range of 1 to  $m$ . In the implemented PIP, the redundancy coefficient  $\eta$  (Section. IV-F) that triggers the start of the confirmation phase is 2. The guard interval is set to 3-bit slots for sequential methods [16]. The presented results are average over 100 simulation runs. In each run, the IDs of  $n$  tags are randomly selected from 1 to  $m$ .

**Baselines:** We compare PIP with the following three baseline identification protocols.

- Q protocol [3]: the *standard sequential* identification protocol adopted in EPC C1G2 [3], in which tags sequentially transmit ID one after another according to dynamic frame slotted ALOHA.
- BIC [23]: the *state-of-the-art sequential* information collection protocol in RFID system, where the reader uses a hash function to allocate a unique time slot to each tag.
- CDMA [12]: a conventional CDMA-based *parallel* method in RFID systems where every tag spreads and transmits its ID by Gold code [19].

### B. Simulation Performance

We first evaluate the performance of PIP with system size  $m$  varying from 100 to 1,000,000 while keeping the number of interrogated tags  $n$  as 100. The identification delay with different protocols are shown in Fig. 6. We can see that PIP outperforms Q, CDMA, and BIC noticeably. For example, In the case with  $m = 10,000$ , the identification delay for Q, CDMA, and BIC is 5,100 bit slots, 4,480 bit slots, and 3,400 bit slots, respectively, while PIP only costs 2,400 bit slots to accomplish the identification process. Specifically, compared

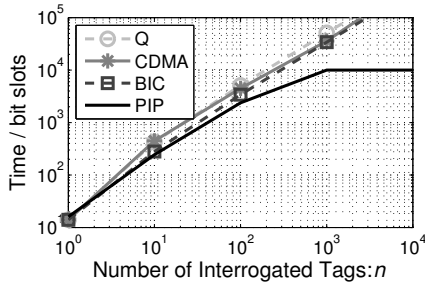


Fig. 7. Comparison on four protocols when  $n$  varying and  $m = 10000$ .

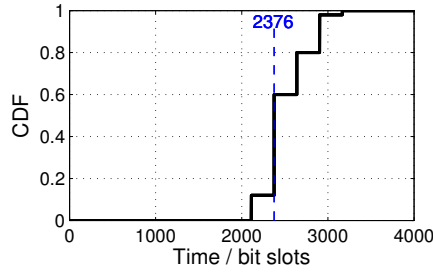


Fig. 8. CDF on the variance of identification delay using PIP when  $n = 100$  and  $m = 10000$ .

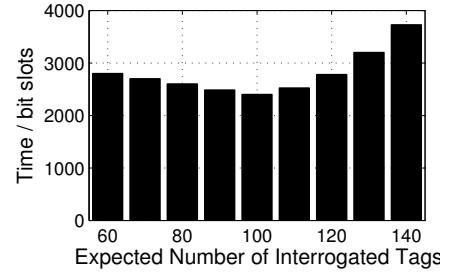


Fig. 9. Impact of estimated  $\hat{n}$  using PIP when  $n = 100$  and  $m = 10000$ .

with the standard Q protocol, PIP reduces the identification delay by at least 50% in all the investigated cases.

Figure 7 shows the identification delay of the four protocols when  $n$  varies from 1 to 10,000 while keeping  $m$  at 10,000. When  $n = 1$ , all protocols cost the similar time to accomplish the identification process, which is intuitive. When  $n$  varies from 10 to 10,000, PIP outperforms all other protocols because the advantage of parallel identification. Note that when  $n$  is large ( $n = 1,000$  and  $n = 10,000$ ), the identification delay of PIP keeps at 10,000 bit slots. This is because that the identification delay with PIP is upper bounded by  $m$ , in which case each tag utilizes a  $m$ -bit string with a single '1' (i.e.,  $L = m$  and  $K = 1$ ) as its encoded string. Figure 7 demonstrates PIP outperforms the other three methods in all investigated  $n$ , especially in large-scale.

Next we investigate the distribution of the identification delay obtained with PIP, which sheds light on whether the performance of PIP is reliable. We consider a system setting with  $n = 100$  and  $m = 10,000$ , in which case the optimal  $L$  is calculated as 264, and  $r$  is calculated as 9. As a result, the expected identification delay is  $264 \times 9 = 2,376$  bit slots. The CDF of the identification delay obtained with simulation is shown in Fig. 8. We can see that the identification delay clusters around 2,376 bit slots in most cases, which indicates a small variance in the performance of PIP. We then extend our investigation by running the simulation with different  $m$  and  $n$ . The results show that the simulated time consumption of PIP is within the range of  $Lr \pm 30\%$  bit slots.

The estimated number of interrogated tags  $\hat{n}$  is required for the reader to identify the optimal settings of PIP. However, the estimated  $\hat{n}$  may deviate from the ground truth  $n$ , and thus we evaluate the robustness of PIP with regard to the estimation error. Figure 9 shows the results with  $m = 10,000$ ,  $n = 100$ , and  $\hat{n}$  varies from 60 to 140. Not surprisingly, we can see that the shortest identification delay is achieved when the estimation is accurate i.e.,  $\hat{n} = n = 100$ . Furthermore, although a longer identification delay is resultant as  $\hat{n}$  deviates from the ground truth, the delay increase is small, e.g.,  $< 5\%$  when  $90 \leq \hat{n} \leq 110$ , which shows PIP is tolerable to the estimation error in the number of interrogated tags.

### C. Performance Insights

The above results demonstrate a high efficiency of the proposed PIP. In the next, we investigate in detail the impacts

of the two critical components of PIP, i.e., the  $L$ - $K$  code and the randomized mapping scheme, on its overall performance. We adopt a smaller scale simulation setting ( $m = 100$ ) to obtain the pattern more clearly. We modify PIP by substituting the  $L$ - $K$  code with the conventional binary code to investigate the efficiency of  $L$ - $K$  code. In this case, to uniquely identify individual tags, a binary string of length  $\lceil B = \log_2 m \rceil$  is required. As a result, every tag encodes the mapped ID into a  $B$ -bit binary segment instead of the  $L$ - $K$  code.

Figure 10 compares the identification delay obtained with  $L$ - $K$  code and the traditional binary code, with  $n$  varying from 1 to 7. We can see that the  $L$ - $K$  outperforms the binary code in all explored cases. Furthermore, it can be observed that the identification delay increases linearly with  $L$ - $K$  code, while it increases exponentially with the binary code. When  $n = 4$ , PIP with binary code demands 819 bit slots to resolve 4 IDs, and when  $n$  increases further, the binary code based PIP may even fail to resolve the ID information. This is because the collision of multiple binary codes has too small segment length ( $B = 7$  in this case) and too many candidate IDs (e.g., when  $n = 5$  and there are  $K' = 6$  '1's in one segment, the number of candidate IDs are  $2^6 = 64$ ). However, according to the equations in Section V,  $L$ - $K$  code has  $L = 17$  and  $K = 2$ . If  $K' = 6$ , the number of candidate is  $\binom{6}{2} = 15$ , which is much smaller than 64. From the analysis and simulation, we find that binary code is not suitable for the parallel identification framework, and thus we adopt  $L$ - $K$  code in PIP.

To investigate the efficiency of the randomized mapping scheme, we implement PIP with shift mapping and enhanced shift mapping schemes mentioned in Section IV-D. Figure 11 presents the identification delay that PIP performs with different mapping schemes when  $n$  varies from 1 to 7. It is clear that randomized mapping scheme achieves the best performance among the three mapping schemes. For example, when  $n = 7$ , PIP with randomized mapping costs about 90 bit slots for identification, while enhanced shift mapping utilizes 149 bit slots, and shift mapping consumes 205 bit slots. This is because both shift mapping and enhanced shift mapping have the obvious equal-difference pattern, which leads to several false IDs appearing with high probability. To filter such false IDs through the intersection operation, the identification delay is prolonged. By contrast, the false IDs generated with the randomized mapping scheme are highly random and thus facilitates the identification process.



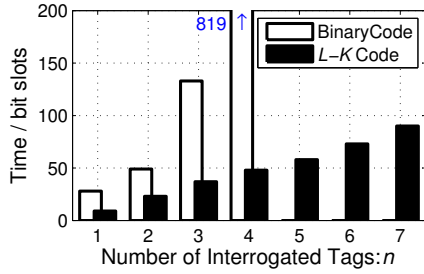


Fig. 10. Impact of coding scheme in PIP when  $n$  varying and  $m = 100$ .

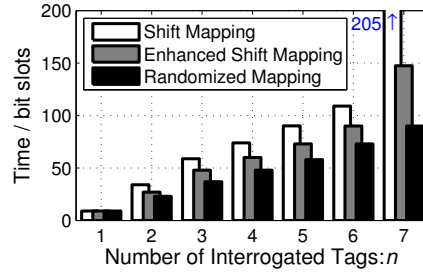


Fig. 11. Impact of mapping scheme in PIP when  $n$  varying and  $m = 100$ .

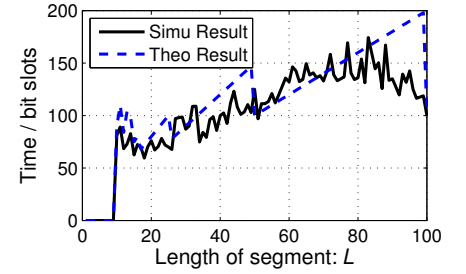


Fig. 12. Comparison on theoretical and simulated results when  $L$  varying,  $m = 100$  and  $n = 5$ .

Then we verify the effectiveness of the proposed method in Section V on determining the optimal  $\langle L, K \rangle$ . Note that we only need to verify the optimality of  $L$  because  $K$  can be determined with  $L$ . Set  $m = 100$ ,  $n = 5$ , we search the identification delay results of  $L$  from its lower bound 9 (obtained by Eq. (5)) to  $m$ . Figure 12 shows the results obtained by the simulation and the theoretical calculation. In the theoretical calculation, since both  $L$  and  $r$  are integers, the resulted curve is sawtooth-shape. On the other hand, the simulated result is averaged over 100 runs, and thus the resultant curve is relatively smooth. We can find the simulated results and the theoretical calculation match each other greatly. Specifically, both the two results indicate that the optimal  $L$  can be found when  $L = 17$ , with the corresponding identification delay around 68 bit slots.

## VIII. CONCLUSION

In this paper, we design a parallel identification protocol to accelerate the identification delay in RFID systems. In order to realize this protocol, we develop a novel  $L$ - $K$  code scheme and a randomized mapping scheme. Taking advantage of these two schemes, tags transmit their mapped-and-encoded IDs concurrently, and the reader can resolve the original IDs from the collision. Theoretical analysis on the optimal values of  $L, K$ , and the expectation of identification delay are derived. Performance evaluation shows that PIP outperforms the existing identification methods on delay.

There are two avenues for our future work. First, improve PIP in the multi-reader RFID systems. Second, extend  $L$ - $K$  code to other wireless communication applications.

## ACKNOWLEDGMENT

This research was supported by Multi-plAtform Game Innovation Centre (MAGIC), funded by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority and Singapore-MIT International Design Center IDG31000101, China NSF grants No. 61303202, No. 61073158, No. 61100210, STCSM Project No. 12dz1507400, No. 13511507800, and the Shanghai Recruitment Program of Global Experts.

## REFERENCES

- [1] M. Buettner, R. Prasad, A. Sample, D. Yeager, B. Greenstein, J. R. Smith, and D. Wetherall. RFID sensor networks with the Intel WISP. In *ACM SenSys*, 2008.
- [2] C. De Vleeschouwer, J.-F. Delaigle, and B. Macq. Circular interpretation of bijective transformations in lossless watermarking for media asset management. *IEEE Transactions on Multimedia*, 5(1):97–105, 2003.
- [3] EPCglobal. Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz.
- [4] K. Finkenzeller. *RFID handbook*. John Wiley & Sons Chichester, UK.
- [5] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman. Challenge: ultra-low-power energy-harvesting active networked tags (EnHANTs). In *ACM MobiCom*, 2009.
- [6] D. K. Klair, K.-W. Chin, and R. Raad. A survey and tutorial of RFID anti-collision protocols. *IEEE Communications Surveys and Tutorials*, 12(3):400–421, 2010.
- [7] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. In *ACM MobiCom*, 2006.
- [8] M. Kodialam, T. Nandagopal, and W. C. Lau. Anonymous tracking using RFID tags. In *IEEE INFOCOM*, 2007.
- [9] T. F. La Porta, G. Maselli, and C. Petrioli. Anticollision protocols for single-reader RFID systems: Temporal analysis and optimization. *IEEE Transactions on Mobile Computing (TMC)*, 10(2):267–279, 2011.
- [10] S. Liu, L. Kang, L. Chen, and L. Ni. Distributed incomplete pattern matching via a novel weighted bloom filter. In *IEEE ICDCS*, 2012.
- [11] G. Marsaglia. The structure of linear congruential sequences. *Applications of number theory to numerical analysis*, pages 249–285, 1972.
- [12] C. Mutti and C. Floerkemeier. CDMA-based RFID systems in dense scenarios: Concepts and challenges. In *IEEE International Conference on RFID*, 2008.
- [13] J. Myung and W. Lee. Adaptive splitting protocols for RFID tag collision arbitration. In *ACM MobiHoc*, 2006.
- [14] C. Qian, H. Ngan, Y. Liu, and L. M. Ni. Cardinality estimation for large-scale RFID systems. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 22(9):1441–1454, 2011.
- [15] Y. Qiao, S. Chen, T. Li, and S. Chen. Energy-efficient polling protocols in RFID systems. In *ACM MobiCom*, 2011.
- [16] M. Shahzad and A. X. Liu. Every bit counts: fast and scalable RFID estimation. In *ACM MobiCom*, 2012.
- [17] W. A. Stein. *Elementary Number Theory: Primes, Congruences, and Secrets*. Springer, 2009.
- [18] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and reliable low-power backscatter networks. *ACM SIGCOMM*, 2012.
- [19] P. Wang, A. Hu, and W. Pei. The Design of Anti-collision Mechanism of UHF RFID System based on CDMA. In *IEEE APCCAS*, 2006.
- [20] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen. Efficient tag identification in mobile RFID systems. In *IEEE INFOCOM*, 2010.
- [21] L. Yang, J. Cao, W. Zhu, and S. Tang. A hybrid method for achieving high accuracy and efficiency in object tracking using passive RFID. In *IEEE PerCom*, 2012.
- [22] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu. Season: Shelving interference and joint identification in large-scale RFID systems. In *IEEE INFOCOM*, 2011.
- [23] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen. A time-efficient information collection protocol for large-scale RFID systems. In *IEEE INFOCOM*, 2012.
- [24] D. Zanetti, B. Danev, et al. Physical-layer identification of UHF RFID tags. In *ACM MobiCom*, 2010.
- [25] Y. Zheng and M. Li. ZOE: Fast cardinality estimation for large-scale RFID systems. In *IEEE INFOCOM*, 2013.
- [26] Z. Zhou, H. Gupta, S. R. Das, and X. Zhu. Slotted scheduled tag access in multi-reader RFID systems. In *IEEE ICNP*, 2007.