Hindawi
Security and Communication Networks
Volume 2017, Article ID 8356983, 9 pages
https://doi.org/10.1155/2017/8356983

Research Article

Time-Efficient Cloning Attacks Identification in Large-Scale RFID Systems

Ju-min Zhao,[1] Ding Feng,[2] Deng-ao Li,[1] Wei Gong,[3] Hao-xiang Liu,[4] and Shi-min Huo[1]

[1] *Taiyuan University of Technology, Taiyuan, China*
[2] *Taiyuan Normal University, Taiyuan, China*
[3] *Tsinghua University, Beijing, China*
[4] *Hong Kong University of Science and Technology, New Territories, Hong Kong*

Correspondence should be addressed to Ju-min Zhao; zhaojumin@tyut.edu.cn

Received 14 July 2016; Revised 26 January 2017; Accepted 8 March 2017; Published 27 April 2017

Academic Editor: Kai Rannenberg

Radio Frequency Identification (RFID) is an emerging technology for electronic labeling of objects for the purpose of automatically identifying, categorizing, locating, and tracking the objects. But in their current form RFID systems are susceptible to cloning attacks that seriously threaten RFID applications but are hard to prevent. Existing protocols aimed at detecting whether there are cloning attacks in single-reader RFID systems. In this paper, we investigate the cloning attacks identification in the multireader scenario and first propose a time-efficient protocol, called the time-efficient Cloning Attacks Identification Protocol (CAIP) to identify all cloned tags in multireaders RFID systems. We evaluate the performance of CAIP through extensive simulations. The results show that CAIP can identify all the cloned tags in large-scale RFID systems fairly fast with required accuracy.

1. Introduction

Radio Frequency Identification (RFID) systems are becoming ubiquitously available in varieties of applications such as inventory control and object tracking. In a large RFID system, each tag with a unique identification (ID) number is attached to an object [1, 2]. The reader can use the ID to search the information of the object and track it [3]. If the IDs and the information of some tags are replicated by the attackers, the cloned tags can be produced. Cloning attack is a grave threat to RFID systems and has attracted wide attention due to its practical importance. For example, RFID tags are the labels of the items in warehouse; since the cloned tags behave exactly the same as genuine tags, counterfeit products can be injected into legal items, causing financial losses [4]. Such problem also appears in applications of healthcare, military, logistics, and so forth [5]. In this case, we can not validate the quality or authenticity of tagged objects.

Aiming at solving the security problem in RFID system, a lot of researchers have invested much vigor. Many international standards have been proposed, such as ISO (International Organization for Standardization) 29167, which can effectively address security protection problem. However, these standards are designed for tags, where the tags could perform security mechanism, whereas this may boost the computation burden of the tags. Thus ISO 29167 standards do not fit the large RFID system. We need to find a more useful approach to settle this problem.

These threats can not be addressed by improving physical architecture that protects the genuine tags from being replicated [6]. These schemes aim at using cryptography and encryption to make tags harder to clone, which are the most intuitive approaches. But they require additional hardware resources and key management strategies [7], which is infeasible for low-cost RFID tags. Though the research community can provide these incremental improvements, it is still not practical to replace or upgrade off-the-shelf tags, since there are already more than 30 billion RFID tags produced globally in 2013 [8]. There is a more promising scheme that aims at verifying tag behaviors against predefined attributes such as the tags location [9, 10]. Although additional hardware is not required, it may leak sensitive information of the items, which is not expected. There are also many prevention protocols proposed in related work, but most of them focus on cloned

tags detection, such as [11]. That is to say, they can only detect whether there are cloning attacks or not. For the power-limited RFID tags, the operational communication distance is very limited. Even for the active tags, the communication distance is only on the order of 100 feet [12]. Hence, in large-scale RFID systems multiple readers are requisite to ensuring the coverage of the region. However, these existing protocols are not suitable in multireader scenario. In this paper, we investigate the cloning attacks identification in the multireader scenario [13]. In many cases, time efficiency and reliability are the most important performance criterions for the solutions. Based on this, we propose a time-efficient protocol, called the time-efficient Cloning Attacks Identification Protocol (CAIP) to identify cloned tags in large RFID systems with multiple readers. To avoid leaking the sensitive information, we do not broadcast the tag ID in our protocol. We use the constructed Bloom filter to efficiently identify the tags in the communication range of one reader (called wanted tags). Then the reader uses multiple hash functions to arrange a unique time slot for each wanted tag. These tags reply to the reader in their own slot. In this way, the protocol execution time is drastically reduced.

Taking the first step toward cloning attack identification in multireader environment, the paper has the following contributions:

(i) This paper proposes CAIP, a pioneer cloning attacks identification scheme. CAIP does not require tag IDs as a priori, which can secure privacy-sensitive applications in large-scale RFID systems.

(ii) We make extensive use of Bloom filter and multiple hash functions to improve the time efficiency. Apart from this, we also exploit the physical layer information to further reduce the operation time.

(iii) CAIPs identification accuracy and execution time are analyzed theoretically. The analysis results can guide protocol configuration for the tradeoff between them.

(iv) We validate the performance of CAIP through extensive simulations. The results show that CAIP can identify all the cloned tags in large-scale RFID systems fairly fast with required accuracy.

The remainder of this paper is organized as follows. We discuss the related work in Section 2. Section 3 gives the system model and problem statement. The detailed design of CAIP is presented in Section 4. The evaluation of our scheme is exhibited in Section 5. We conclude this paper in Section 6. In Acknowledgments, we give all organizations that funded our research.

2. Related Work

In the existing work, a large body of research has been conducted on various issues in RFID systems, such as information collection, RFID identification, cardinality estimation, and item monitoring.

The existing protocols can be classified into three broad categories: Aloha-based [14–16], tree-based [17], and hybrid [18]. In Aloha-based protocols, the reader broadcasts a query request to the tags in its query range. On receiving the query request, each tag chooses a time slot, with a certain probability, to transmit its information. The tags cannot be identified due to tag-tag collisions if more than one tag chooses the same time slot. In tree-based protocols, the reader detects whether collisions occur and divides the tag set into small subsets if there is a collision. The reader repeats the process until no collision occurs. Our proposed protocol is based on frame-slot Aloha protocol.

RFID systems include tags which are attached to objects, RFID readers that read and write data on tags, and back-end systems that store and share data. Cryptographic RFID tags are currently widely available in the HF band, but today there are no cryptographic tags commercially available in the UHF band. Cloning attacks threaten Radio Frequency Identification (RFID) applications but are hard to prevent.

Conventional solutions comprise prevention, authentication, and detection. Most existing prevention protocols use cryptography and encryption to make tags hard to clone [19, 20]. But they require additional hardware resources and key management strategies.

Authentication is a sharp weapon against counterfeit tags that carry valid IDs but forged keys [21]. It includes reader-to-tag and tag-to-reader authentication. Several tag-to-reader authentication protocols have been proposed in [22, 23]. They are most pseudorandom numbers and hash functions. But cloned tags hold not only valid IDs but also valid keys.

Detection measures do not require cryptographic operations from the tags but they make use of visibility to detect cloned tags or changes in the tag ownership. Reference [24] developed a system that essentially detects cloned RFID tags or other changes in tag ownership in an access control application using intrusion detection methods. Many other solutions have been presented [6, 25].

These protocols aim at detecting whether there are cloning attacks and they are not suitable in multireader RFID systems. However, in many cases, we want to identify all the cloned tags. In this paper, we investigate the cloning attacks identification in the multireader scenario and propose a time-efficient protocol, called the time-efficient Cloning Attacks Identification Protocol (CAIP) to identify cloned tags in large RFID systems with multiple readers.

3. System Model and Problem Statement

3.1. System Model. Consider a large-scale RFID system with numerous tags and multiple readers. Every tag carries a unique ID and has the capability of performing certain computations as well as communications. These readers are reasonably deployed and each of them has a communication region. The tags distributed in a readers communication region, called interrogated tags, can communicate with the reader.

We assume that all the RFID readers have access to a back-end server, which stores all the IDs of all tags. All the readers are synchronized by the back-end server and can be logically treated as one. This assumption is necessary and it is also made in [26]. We can get the IDs by updating the database

when the items move into or out of the RFID system. This is what a typical RFID system management procedure will do. Even if the IDs are lost due to a database failure, we can easily retrieve them by operating the ID-collection protocols such as [17]. In addition we assume that cloned and original tags are in the interrogation region of the system at the same time.

Communications between RFID readers and tags adopt time slot, which follows the Reader-Talks-First protocol [2]. At the beginning, the reader transmits a command query to initialize each round of communication. At the same time, the clocks of tags are well synchronized by the signal received from the reader. Then, several tags respond during a subsequent slotted time frame. If no tag responds in a slot, the slot is called empty slot; if only one tag responds, it is called a singleton slot; if more than one tag responds, it is a collision slot. A singleton or collision slot is called a nonempty slot. In many cases, we only need to separate the nonempty slots from the empty ones, where the tags can transmit one-bit short responses (i.e., 0 represents empty and 1 represents nonempty). Otherwise, we need to determine whether a slot is an empty slot, singleton slot, or a collision slot; the tags should transmit a multibit long-response, which is 10 bits in the Philips I-Code system.

Philips I-Code system belongs to ISO 15693 Standards, and I-Code system could offer simultaneous operation in the covered file of the reader. Thus when there exists a clone tag, there would be two same tags in the inventory, one is authentication, and the other is counterfeit. And the same tag has the same ID, which means in the process of hash function these two same tags would acquire the same value. Meanwhile, the same value means these two tags choose the same slot to respond to the reader's inquiry. That is the reason why collision slot appears. Let us consider in reverse, if we identify the conflicting tags, we have recognized the cloned tag.

In this paper, we denote the length of tag slots, used to transmit the 96-bit ID or a segment, as t_{tag}. The lengths of a long-response slot and a short-response are denoted as t_l and t_s, respectively. We adopt the parameters of Philips I-Code system in our numerical examples and the simulation. With respect to the waiting time between the transmissions, $t_{\text{tag}} = 2.4$ ms, $t_l = 0.8$ ms, and $t_s = 0.4$ ms.

3.2. Problem Statement. The problem is to design an efficient protocol to identify all the cloned tags with minimum execution time in a multireader RFID system. In the rest of the paper, it is also termed as the multireader cloned tags identification problem.

The multireader cloned tags identification problem is quite different from cloned tags detection in the single-reader scenario. In the single-reader system, all tags communicate with one reader. However, in a multireader RFID system, each reader has its own communication region. Although each reader has access to the IDs of all tags, it has no knowledge about the tags in its interrogation region due to the mobility of the tags. Hence, we should first find the wanted tags, which makes the multireader cloned tags identification problem complicated and challenging to be addressed.

The cloned tags hold the information of genuine tags including the ID, so the collisions can not be addressed by arbitrating channel access among tags. That is to say the tags with the same ID will respond in the same slot. Based on this observation, we can assign a singleton slot for each tag; if the expected singleton slots turn to collision slots, we can determine that the tag is cloned. Therefore, we have to determine whether a slot is singleton or collision, which means that the tags must transmit long-response to the reader.

We assume that there are n tags in a large-scale RFID system N. Let $M \subseteq N$ represent the set of tags in a readers interrogation region, and m is the number of M. Let $S \subseteq M$ is the set of attacked tags and s presents the number of S. Our aim is to identify S. The reader need first identify the interrogated tags M in order to find the cloned tags set S. To fast and reliably identify these interrogated tags, we introduce Bloom filter. Then, we use multiple hash functions to allocate a singleton slot for each interrogated tag. If one tag is cloned (i.e., the attacked tag), the expected singleton slot, corresponding to the attacked tag, will turn to a collision slot.

4. Time-Efficient Cloning Attacks Identification Protocol (CAIP)

To address the problem, the reader first identifies the interrogated tags through Bloom filter. Then, we use multiple hash functions to allocate a singleton slot for each interrogated tag. By finding the tags corresponding to collision slots, the attacked tags set S can be determined. In this section, we present the time-efficient Cloning Attacks Identification Protocol (CAIP) in detail. Section 4.1 gives the procedure of identifying the interrogated tags, which takes advantage of the synchronized physical layer transmissions to distributively construct the desired Bloom filter. Sections 4.2 and 4.3 show how to arrange time slots for tags using a hash function and multiple hash functions, respectively. The procedure of identifying all the cloned tags is presented in Section 4.4. Finally, we give the execution time analysis in Section 4.5.

4.1. Interrogated Tag Identification. Bloom filter is a simple space-efficient probabilistic data structure for representing a set and supporting membership queries [27]. Hence, if the tag set M can be transmitted to the reader in this form, the overhead for identifying interrogated tags could be significantly reduced [28]. It is a challenge to construct the Bloom filter in the case that the reader has no idea about M. Another challenge is how to improve the utilization rate of time slots in order to reduce the execution time.

In this phase, the reader first broadcasts an operation code, which contains two parameters w and k. w represents the size of the Bloom filter and k is the number of the hash functions. That is to say, we use k hash functions to construct the Bloom filter.

We denote the k hash functions as $h_1, h_2, \ldots, h_k$, each with the range of $\{0, 1, \ldots, w-1\}$. When receiving the operation code, each tag in M generates a w-bit array, which is initialized to 0. With k hash functions using each tags ID as the seed, each tag sets the positions $h_1(\text{ID}), h_2(\text{ID}), \ldots, h_k(\text{ID})$ in

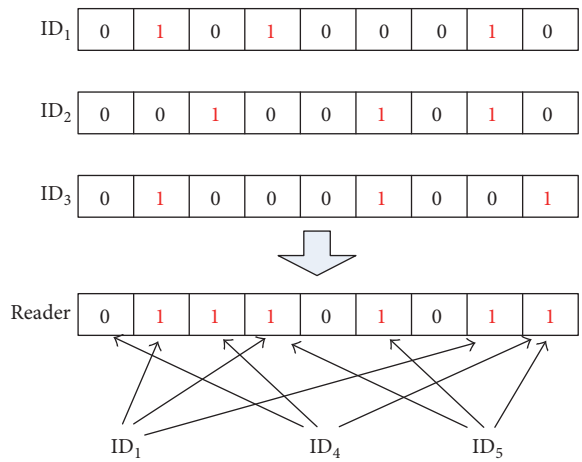

Figure 1: We use a simple example, where each tag uses three hash functions to construct the Bloom filter, to illustrate the procedures of the interrogated tag identification.

the array to 1. We call this array Bloom filter vector. Then, all the tags in M simultaneously transmit their own Bloom filter vector. In the physical layer, an idle carrier represents 0 and a busy carrier represents 1 [29]. The reader receives the superposition of all the Bloom filter vectors transmitted from the tags and generates a new w-bit array v.

We use a simple example, where $k = 3$ and $w = 9$, to illustrate the procedures of the interrogated tag identification, as shown in Figure 1. Once constructing the v, the reader hashes each tag in set N to k positions $h_1(\mathrm{ID}), h_2(\mathrm{ID}), \ldots, h_k(\mathrm{ID})$ in v. For the tags in set M, all the positions should be 1 (such as tag with ID_1). If any of them are 0, the tag is not in M (such as the tag with ID_4). However, according to the property of Bloom filter, a tag may be not in M, but all the k according positions are 1 (such as the tag with ID_5). This is false positive. We denote the tag set retrieved by the reader as $\widetilde{M}$. The expected cardinality of false positive tags can be represented as $p \times (n - m)$, where p is the probability of false positives.

In the following, we show how to determine the parameters w and k. The false positive probability p can be represented as

$$p = \left[1 - \left(1 - \frac{1}{w}\right)^{kn}\right]^k \approx \left(1 - e^{-kn/w}\right)^k. \quad (1)$$

With a given m and p, the length of the Bloom filter vector can be written as $w = -(n \times \ln p)/(\ln 2)^2$, and the optional value of k is $k = (w/n) \ln 2$.

4.2. Assigning Tags to Time Slots Using a Hash Function. In the above section, the reader attains the set $\widetilde{M}$. It can start to identify all the cloned tags from $\widetilde{M}$. Intuitively, we can broadcast each tag ID and wait for its long-response. If collisions occur, this indicates that this tag is attacked. Otherwise, the tag is not attacked. We call this protocol Polling Identification Protocol (PIP). However, it is not efficient and can leak the sensitive information. In our protocol, we do not broadcast the tag IDs or indices. At first, we show how to arrange time slots for tags using a hash function.

At the beginning of a phase, the reader first broadcasts the query command to all the tags in its communication range, which contains the random number r and the frame size f, where r is used by the hash function and it is different in each phase. Considering an arbitrary phase, we assume that there are m' tags not identified. That is to say, we do not know whether the m' tags are attacked or not. Therefore, we only consider assigning time slots for these tags. Clearly, $m' = m + p \times (n - m)$ in the first phase.

We know that when the frame size $f = m'$, the slots utilization is the fullest. The probability of slots utilization is $P_1 = (1 - 1/m')^{m'-1} \approx e^{(m'-1)/m'} \approx e^{-1} \approx 36.8\%$. Hence, in each phase, the reader sets $f = m'$. Before the reader transmits a request, they have to determine which tags should respond in this phase and which slots in the frame they should be assigned to. The reader should avoid assigning more than one tag to a slot in order to reduce wasted slots. The reader selects a random number r and maps the IDs to the slots through the hash function. Then they know which slots are singleton slots (i.e., the useful slots) and which are not singleton (the wasted slots). The reader constructs an n'-bit indicator vector, where each bit corresponds to slot in the current frame. If only one tag corresponds to a slot, the representative bit in the vector is set as 1; otherwise, it is set as 0.

In each phase, the reader broadcasts a query consisting of the indicator vector along with f and r. If the vector is too long, the reader divides it into 96-bit segments, which is equal to the length of the tag ID, and transmits each of them in t_{id}.

Once receiving the query, the tags know the index i of the slot it mapped to using the same hash function and r. Each of them knows whether its slot is useful or not by examining the ith bit in the indicator vector. If the ith bit is 1, the tag will transmit a long-response in the ith slot in the current frame. Otherwise, it will keep silence. It should be noted that the tag can receive the required segment instead of the whole indicator vector since it knows which segment is the one it

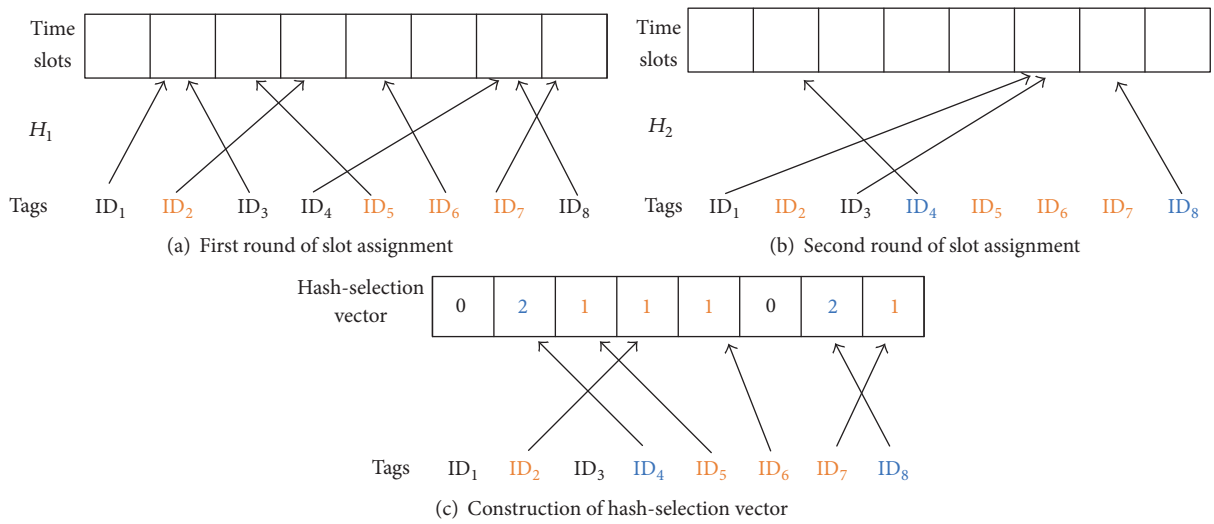

FIGURE 2: An example, where $K = 2$, is presented to illustrate the process of the construction of hash-selection vector.

looks for. The tag can keep stand-by to save energy until it receives its segment.

We can see that only 36.8% of a frame is useful slots. That is to say, in each frame, 63.2% of all the slots are wasted. This motivates us to propose the more efficient protocol CAIP.

4.3. Assigning Tags to Slots Using Multiple Hash Functions. In this section, we explain how to arrange a singleton time slot for each of the tags with K hash functions.

It is different from the single-hash protocol presented above. In each phase, we use K hash functions to map each tag to a singleton slot. It has K rounds. In the first round, we use H_1 to assign tags to slots. If only one tag maps to the slot i, the tag is removed from being further considered in the remaining rounds and the slot is labeled as occupied slot. In the end of the first round, all the nonsingleton slots are labeled as unoccupied slots. The remaining tags and the unoccupied slots will participate in the second round. The second round is similar to the first round; all the remaining tags are mapped to the unoccupied slots using hash function H_2. After this round, the unassigned tags and unoccupied slots will take part in the third round.

Repeat this process using the remaining hash functions for K rounds. After K rounds, the reader knows the subset of tags assigned to singleton slots and the hash functions each of these tags use. If a slot is still remaining unoccupied after K rounds, it will participate in the next frame.

In Figure 2, an example, where $K = 2$, is presented to illustrate the process of the construction of hash-selection vector. In Figure 2(a), there are four tags (ID_2, ID_5, ID_6, ID_7) assigned to singleton slots with the hash function $H1$ in the first round. In Figure 2(b), ID_4 and ID_8 are mapped to singleton slots in the second round using H_2. Finally, the reader constructs the hash-selection vector V according to the two rounds as shown in Figure 2(c).

4.4. Cloning Attacks Identification. In this section, we present the efficient Cloning Attacks Identification Protocol (CAIP). At the beginning of the CAIP, we first find out the interrogated tag set M from N. Then, we identify all the cloned tags based on multihash functions, which had two phases.

In the first phase, before transmitting the query command, the reader determines which tags are mapped to which slots as mentioned in previous sections. Then, it constructs an $[m + p \times (n - m)]$-element hash function vector V with multihash functions. Each element of V corresponds to a slot in the current frame at the same index location. If a tag is assigned to a singleton slot using the jth hash function, the reader sets the corresponding element to be j. The length of an element is $\log_2 K + 1$ bits. After K hash functions, if one slot is not occupied successfully, the reader will set the corresponding element in the hash-selection vector to zero, and if a tag is not assigned to a singleton slot, the tag will participate in the next frame.

The second phase includes several rounds. At beginning of each round, the reader broadcasts the frame size f, a random number r, and the hash-selection vector V to the tags in its communication region. If the hash-selection vector V is too long to transmit, it is divided into 96-bit segments and each segment is transmitted in a time slot t_{id}.

The tags will respond according to V and we can determine which tags are cloned by examining the corresponding slots. Recall that we allocate a singleton slot for each tag; the corresponding slot will turn to a collision slot if a tag is cloned. Once the tags received the request sent out by the reader, each tag uses the same K hash functions one by one to find out the K representative elements in V. If a tag is mapped to an element with the value of j with the H_j hash function, this tag knows it is assigned a singleton slot. Then the tag calculates how many nonzero elements appear before its indicator elements in V. Each nonzero element represents a tag that is scheduled to respond in the corresponding time slot. If there are q nonzero elements before its indicator elements, the tag should respond in the $(q + 1)$th time slots, and it will not consider the remaining hash functions. If

a tag does not find a singleton slot after using all the K hash functions, it will keep silence in this round and participate in the next round. The above rounds are repeated until all tags are checked.

4.5. Execution Time Analysis. The overall execution time of CAIP is equal to the sum of the time taken by the reader to identify the interrogated tags and identify all the cloned tags. Within the interrogated tag identification, the reader broadcasts a request command; then the tags transmit their w-bit Bloom filter vectors simultaneously. Neglecting the transmission time, this time for identifying the interrogated tags can be calculated as

$$T_1 = w \times t_s = -\frac{n \times \ln p}{(\ln 2)^2} \times t_s. \tag{2}$$

To compute the expected execution time of cloned tags identification, we need to determine how many rounds an arbitrary tag is expected to participate in. Consider an arbitrary tag t and an arbitrary round that t participates in. Similar to the previous section, the frame size of each round is m', which is equal to the number of tags that participate in this round. Let P_j be the probability that tag t is assigned to a singleton slot after the first j hash functions. When tag t is mapped to an element of V and assigned to a singleton slot successfully, it will calculate the amount of nonzero elements before its indicator element to determine in which slot it will respond. After this, the tag t will not participate in the remaining rounds.

We know that $P_1 = e^{-1} = 36.8\%$; now we continue to calculate P_j ($j > 1$). After the first $j-1$ hash functions are used, there are two cases for the tags and the slots. The first case is that tag t has been assigned to a singleton slot successfully and the probability is P_{j-1}. The second case is that tag t has not been assigned to any singleton slot and the probability is $1 - P_{i-1}$. These tags will participate in the jth round. Since the number of tags is equal to the number of slots in each round and the slot assignment is one-to-one mapping, the probability for a slot to stay unoccupied after $j-1$ hash functions is $1 - P_{j-1}$. In the jth round, tag j is mapped to an unoccupied slot with probability $1-P_{j-1}$. For each of the other $m'-1$ tags, it participates in the jth round with the probability $1 - P_{j-1}$, and it is mapped to the same slot as tag t does with probability $1/m'$. Therefore, the probability p' for tag t to be the mapped to a singleton slot in the jth round can be written as

$$\begin{aligned} p' &= \left(1 - P_{j-1}\right)\left(1 - \left(1 - P_{j-1}\right)\frac{1}{m'}\right)^{m'-1} \\ &\approx \left(1 - P_{j-1}\right) e^{-(1-P_{j-1})}. \end{aligned} \tag{3}$$

Now, we can derive a recursive formula for P_i according the above analysis. P_j should be the sum of the two cases, which are the probability for a tag to be assigned to a singleton slot before by one of the first $j-1$ hash functions and the probability to be assigned to a singleton slot by the jth hash function.

$$\begin{aligned} R1P_j &= P_{j-1} + p'\left(1 - P_{j-1}\right) \\ &= P_{j-1} + \left(1 - P_{j-1}\right)^2 e^{-(1-P_{j-1})}. \end{aligned} \tag{4}$$

Then, the expected number of rounds where tag t participates in $E(K)$ can be calculated as

$$E(K) = \sum_{j=1}^{K} \left(j \times \left(1 - P_K\right)^{j-1} \times P_K\right) = \frac{1}{P_K}. \tag{5}$$

We can easily compute the total expected time in the second phase. The expected number of rounds that an arbitrary tag participates in is $1/P_K$; that is, each tag needs $1/P_K$ elements to be assigned to a slot. Recall that the length of an element is $\log_2(K+1)$ bits; hence the expected length of required elements should be $\log_2(K+1)/P_K$. The expected cardinality of tag set $\widetilde{M}$ determined in the interrogated tags identification phase is $m + p \times (n-m)$. After being divided into 96-fit segments, the expected time for the whole hash-selection vector is $((m+p\times(n-m))\log_2(K+1)/96P_K)t_{\mathrm{id}}$. Since the tags are one-to-one responding and they must respond to multiple bits to determine whether it is singleton slot or collision slot, the total time for all tags to respond is expected to be $(m + p \times (n-m))t_l$. We can compute the expected time in the second phase as

$$\begin{aligned} T_2 &= \frac{\left(m + p \times (n-m)\right)\log_2(K+1)}{96P_K} t_{\mathrm{id}} \\ &\quad + \left(m + p \times (n-m)\right) t_l \\ &= \left[\frac{\log_2(K+1)}{96P_K} t_{\mathrm{id}} + t_l\right]\left[m + p \times (n-m)\right]. \end{aligned} \tag{6}$$

Based on the above analysis, the expected execution time of CAIP, denoted as T, can be computed as follows:

$$\begin{aligned} T &= -\frac{n \times \ln p}{(\ln 2)^2} \times t_s \\ &\quad + \left[\frac{\log_2(K+1)}{96P_K} t_{\mathrm{id}} + t_l\right]\left[m + p \times (n-m)\right]. \end{aligned} \tag{7}$$

5. Performance Evaluation

In this section, we evaluate the performance of CAIP through numerous simulations. Since CAIP is the first protocol for cloning attacks identification in multireader RFID systems, we have no comparison when conducting simulations. To show the good performance of our protocol, we compare time with GREAT, which is proposed in [11] to detect cloning attacks. In GREAT, the tags are expected to decide when to respond according to their IDs such that tags with the same ID always simultaneously respond. Tags with different IDs could, however, respond either simultaneously or asynchronously. If tags with different IDs respond simultaneously and cause a collision, we are likely to reconcile the collision by further arbitrating access to the channel among them. On the other hand, if a collision is due to responses from

Table 1: Execution time comparison (in seconds) when $n = 10000$ and $m = 0.1n$.

(n, m)	$s = 2$			$s = 10$		
	GREAT	PIP	CAIP	GREAT	PIP	CAIP
(5000, 500)	12.6	16.0	0.6	6.5	16.0	0.6
(10000, 1000)	25.3	32.0	1.3	13.1	32.0	1.3
(15000, 1500)	38.0	48.0	1.8	19.7	48.0	1.8
(20000, 2000)	50.7	64.0	2.5	26.2	64.0	2.5
(25000, 2500)	63.3	80.0	3.1	32.8	80.0	3.1
(30000, 3000)	76.0	96.0	3.7	39.4	96.0	3.7
(35000, 3500)	88.7	112.0	4.4	45.9	112.0	4.4
(40000, 4000)	101.4	128.0	5.0	52.5	128.0	5.0
(45000, 4500)	114.0	144.0	5.6	59.1	144.0	5.6
(50000, 5000)	126.8	160.0	6.3	65.6	160.0	6.3

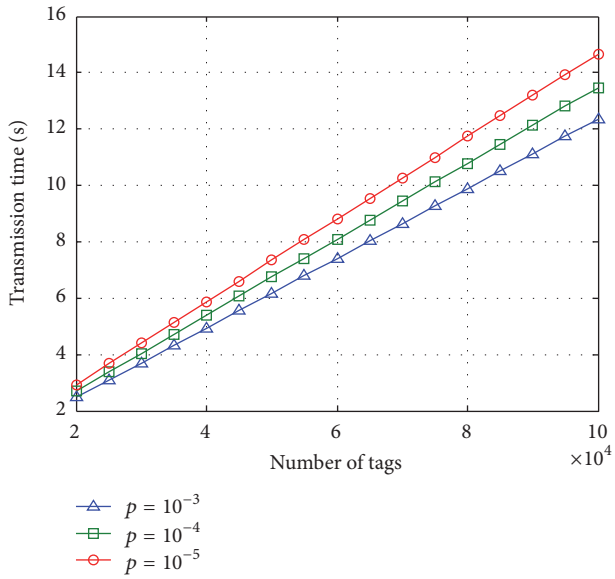

Figure 3: Transmission time with different p when $m = 0.1n$.

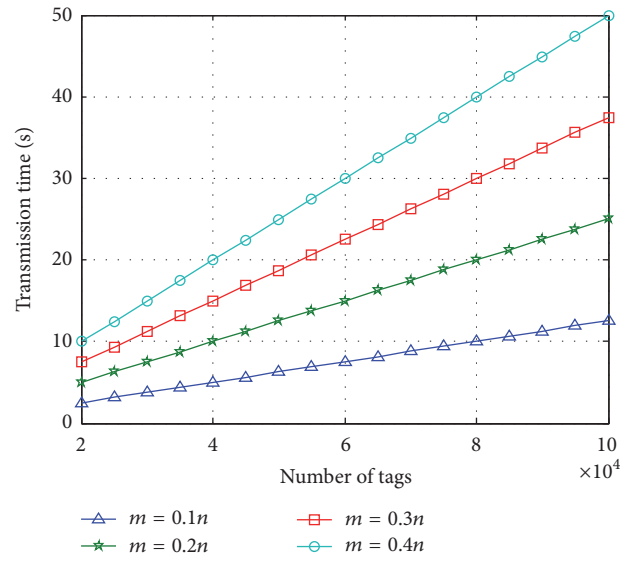

Figure 4: Transmission time with different m when $p = 10^{-3}$.

tags with the same ID, it is hard to reconcile. It is worth noting that GREAT can only detect whether there are cloned tags or not and can not identify all the cloned tags. We also make comparison with Polling Identification Protocol (PIP) presented in Section 5.2, which is used for cloning attacks identification.

5.1. Execution Time under Different Parameters. In our simulations, we set $K = 7$. We first evaluate the performance of CAIP under different values of Bloom filter false positive p ($p = 10^{-3}$, $p = 10^{-4}$, and $p = 10^{-5}$) when $m = 0.1n$, which is shown in Figure 3. It is noted that the execution time decreases with the increasing of p. Hence, in the following comparison with the other protocols we set $p = 10^{-3}$. We also evaluate CAIP with different n, which varies from 20000 to 100000, when $p = 10^{-3}$. For each value of m, we set $m = 0.1n$, $m = 0.2n$, $m = 0.3n$, and $m = 0.4n$, shown in Figure 4. It is shown that the execution time increases with the value of m.

5.2. Performance Comparison. We also performed extensive simulations to compare the performance of our CAIP with the most related work, GREAT and PIP. The two protocols do not consider the multireader environment; that is, the reader has no knowledge about tags located in its interrogation region. Therefore, for the PIP, the reader must check all the tags to identify all the cloned tags, and for GREAT the reader must examine all the tags to detect cloning attacks.

Table 1 illustrates the execution time of GREAT, PIP, and CAIP when $n = 10000$ and $m = 0.1n$. In this simulation, we set the number of cloned tags $s = 2$ and $s = 10$, respectively. It is shown that CAIP outperforms all the other protocols. For example, when $s = 2$, the time taken by CAIP is only 5% of that taken by GREAT, and it is 4% or so of PIP. When $s = 10$, the time taken by CAIP is only 10% of that taken by GREAT, and it is 4% or so of PIP. It is noted that the time taken by GREAT increases when the number of cloned tags s decreases, while it has nothing to do with s for PIP and CAIP, since GREAT is used to detect if there are cloning attacks and PIP and CAIP are proposed for identifying all the cloned tags.

Table 2: Execution time comparison (in seconds) when $n = 10000$ and $m = 0.3n$.

(n, m)	$s = 2$			$s = 10$		
	GREAT	PIP	CAIP	GREAT	PIP	CAIP
(5000, 1500)	12.6	16.0	1.9	6.5	16.0	1.9
(10000, 3000)	25.3	32.0	3.7	13.1	32.0	3.7
(15000, 4500)	38.0	48.0	5.6	19.7	48.0	5.6
(20000, 6000)	50.7	64.0	7.5	26.2	64.0	7.5
(25000, 7500)	63.3	80.0	9.4	32.8	80.0	9.4
(30000, 9000)	76.0	96.0	11.2	39.4	96.0	11.2
(35000, 10500)	88.7	112.0	13.1	45.9	112.0	13.1
(40000, 12000)	101.4	128.0	15.0	52.5	128.0	15.0
(45000, 13500)	114.0	144.0	16.9	59.1	144.0	16.9
(50000, 15000)	126.8	160.0	18.7	65.6	160.0	18.7

Table 3: Execution time comparison (in seconds) when $n = 10000$ and $m = n$.

(n, m)	$s = 2$			$s = 10$		
	GREAT	PIP	CAIP	GREAT	PIP	CAIP
(5000, 5000)	12.6	16.0	6.2	6.5	16.0	6.2
(10000, 10000)	25.3	32.0	12.5	13.1	32.0	12.5
(15000, 15000)	38.0	48.0	18.7	19.7	48.0	18.7
(20000, 20000)	50.7	64.0	24.9	26.2	64.0	24.9
(25000, 25000)	63.3	80.0	31.2	32.8	80.0	31.2
(30000, 30000)	76.0	96.0	37.4	39.4	96.0	37.4
(35000, 35000)	88.7	112.0	43.6	45.9	112.0	43.6
(40000, 40000)	101.4	128.0	49.9	52.5	128.0	49.9
(45000, 45000)	114.0	144.0	56.1	59.1	144.0	56.1
(50000, 50000)	126.8	160.0	62.3	65.6	160.0	62.3

Tables 2 and 3 show the execution time when $m = 0.3n$ and $m = n$, respectively. We observe that CAIP still achieves the highest time efficiency compared with the other protocols. When $s = 2$, $m = 0.3n$, the time taken by CAIP is only 15% of that taken by GREAT, and it is 12% or so of PIP. When $s = 2$, $m = n$, the time taken by CAIP is only 50% of that taken by GREAT and 39% or so of PIP. It should be stressed that our proposed CAIP can identify all the cloned tags, while GREAT can only detect whether there are cloned tags.

6. Conclusion

Cloning attacks seriously threatened RFID applications but are hard to prevent. Existing protocols aimed at detecting whether there are cloning attacks in single-reader RFID systems. In this paper, we investigate the cloning attacks identification in the multireader scenario and first propose CAIP, a pioneer cloning attacks identification scheme in multireaders RFID systems. CAIP does not require tag IDs in one certain reader's region as a priori, which can secure privacy-sensitive applications in large-scale RFID systems. But the implementation of CAIP must be based on the assumption that clone and original tags are in interrogation region of the system at the same time. We make extensive use of Bloom filter and multiple hash functions to improve the time efficiency. Apart from this, we also exploit the physical layer information to further reduce the operation time. We evaluate the performance of CAIP through extensive simulations. The results show that CAIP can identify all cloned tags in large-scale RFID systems fairly fast with required accuracy.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The paper is supported by the General Object of National Natural Science Foundation under Grant 61572346: The Key Technology to Precisely Identify Massive Tags RFID System with Less Delay; International Cooperation Project of Shanxi Province under Grant 2015081009: The Search and Analysis of Popular Categories Based on Energy; International Cooperation Project of Shanxi Province under Grant no. 201603D421012: Research on the Key Technology of GNSS Area Strengthens Information Extraction Based on Crowd Sensing; the General Object of National Natural Science Foundation under Grant 61572347: Resource Optimization in Large-Scale Mobile Crowd Sensing: Theory and Technology.

References

[1] EPC class-1 generation-2 RFID protocol, V. 1. 0, http://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_1_2_0-standard-20080511.pdf.

[2] Philips Semiconductors, I-CODE Smart Label RFID Tags, http://www.semiconductors.philips.com/.

[3] Y. Zheng and M. Li, "Fast tag searching protocol for large-scale RFID systems," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 924–934, 2013.

[4] D. Delen, B. C. Hardgrave, and R. Sharda, "RFID for better supply-chain management through enhanced information visibility," *Production and Operations Management*, vol. 16, no. 5, pp. 613–624, 2007.

[5] B. D. Janz, M. G. Pitts, and R. F. Otondo, "Information systems and health care-II: back to the future with RFID: lessons learned-some old, some new," *Communications of the Association for Information Systems*, vol. 15, no. 1, article 7, 2005.

[6] M. Lehtonen, D. Ostojic, A. Ilic et al., in *Securing RFID systems by detecting tag cloning/Pervasive Computing*, pp. 291–308, Springer, Berlin Heidelberg, Germany, 2009.

[7] S. Spiekermann and S. Evdokimov, "Privacy enhancing technologies for RFID-a critical state-of-the-art report," *IEEE Security and Privacy*, vol. 7, no. 2, pp. 56–62, 2009.

[8] http://www.tldm.org/news4/markofthebeast.htm.

[9] M. Lehtonen, F. Michahelles, and E. Fleisch, "How to detect cloned tags in a reliable way from incomplete RFID traces," in *Proceedings of the IEEE International Conference on RFID (RFID '09)*, pp. 257–264, Atlanta, GA, USA, April 2009.

[10] D. Zanetti, L. Fellmann, and S. Capkun, "Privacy-preserving clone detection for RFID-enabled supply chains," in *4th Annual IEEE International Conference on RFID, RFID 2010*, pp. 37–44, usa, April 2010.

[11] K. Bu, X. Liu, J. Luo, B. Xiao, and G. Wei, "Unreconciled collisions uncover cloning attacks in anonymous RFID systems," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 429–439, 2013.

[12] A. Ruhanen, M. Hanhikorpi, F. Bertuccelli et al., *Sensorenabled RFID Tag Handbook*, IST-2005-033546, BRIDGE, 2008.

[13] D. Feng, *Research on Time-Efficient Cloning Attacks Identification in Large Scale RFID Systems*, Taiyuan University of Technology, 2014.

[14] H. Vogt, "Efficient object identification with passive RFID tags," in *Pervasive Computing*, vol. 2414 of *Lecture Notes in Computer Science*, pp. 98–113, Springer, Berlin, Germany, 2002.

[15] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large RFID system," in *Proceedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '11)*, pp. 277–286, Salt Lake City, Utah, USA, June 2011.

[16] S. Chen, M. Zhang, and B. Xiao, "Efficient information collection protocols for sensor-augmented RFID networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 3101–3109, April 2011.

[17] J. Myung and W. Lee, "Adaptive splitting protocols for RFID tag collision arbitration," in *Proceedings of the ACM Mobile Ad Hoc*, 2006.

[18] 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*, vol. 10, no. 2, pp. 267–279, 2011.

[19] J. Abawajy, "Enhancing RFID tag resistance against cloning attack," in *Proceedings of the 3rd International Conference on Network and System Security (NSS'09)*, pp. 18–23, October 2009.

[20] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm'05)*, pp. 59–66, Greece, September 2005.

[21] C. Tan, B. Sheng, and Q. Li, "Secure and serverless RFID authentication and search protocols," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1400–1407, 2008.

[22] A. Juels, "Minimalist cryptography for low-cost RFID tags," *Security in Communication Networks*, pp. 149–164, 2005.

[23] J. Yang, J. Park, H. Lee, K. Ren, and K. Kim, "Mutual authentication protocol for low-cost RFID," in *Proceedings of the ECRYPT Workshop on RFID and Lightweight Crypto*, 2005.

[24] L. Mirowski and J. Hartnett, "Deckard, a system to detect change of RFID tag ownership," *International Journal of Computer Science and Network Security*, vol. 7, no. 7, pp. 89–98, 2007.

[25] R. Koh, E. W. Schuster, I. Chackrabarti, and A. Bellman, "Securing the Pharmaceutical Supply Chain," White Paper, Auto-ID Labs, Massachusetts Institute of Technology, 2003.

[26] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '10)*, pp. 1–10, September 2010.

[27] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[28] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, "A time-efficient information collection protocol for large-scale RFID systems," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'12)*, pp. 2158–2166, Atlanta, GA, USA, March 2012.

[29] Y. Zheng and M. Li, "P-MTI: physical-layer missing tag identification via compressive sensing," in *Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM '13)*, pp. 917–925, April 2013.