Fast Cloned-Tag Identification Protocols for Large-Scale RFID Systems

Kai Bu, Xuan Liu, Bin Xiao
Department of Computing, The Hong Kong Polytechnic University
Email: {cskbu, csxuanliu, csbxiao}@comp.polyu.edu.hk

***Abstract*—Tag cloning attacks threaten a variety of Radio Frequency Identification (RFID) applications but are hard to prevent. To secure RFID applications that confine tagged objects in the same RFID system, this paper studies the cloned-tag identification problem. Although limited existing work has shed some light on the problem, designing fast cloned-tag identification protocols for applications in large-scale RFID systems is yet not thoroughly investigated. To this end, we propose leveraging broadcast and collisions to identify cloned tags. This approach relieves us from resorting to complex cryptography techniques and time-consuming transmission of tag IDs. Based on this approach, we derive a time lower bound on cloned-tag identification and propose a suite of time-efficient protocols toward approaching the time lower bound. The execution time of our protocol is only 1.4 times the value of the time lower bound, being up to 91% less than that of the existing protocol. The proposed protocols may benefit also RFID applications that distribute tagged objects across multiple places.**

I. Introduction

Tag cloning attacks threaten a variety of Radio Frequency Identification (RFID) applications but are hard to prevent. Launching a tag cloning attack, an attacker compromises genuine tags and produces their replicas, namely *cloned tags* [1]. Cloned tags behave exactly the same as genuine tags and can pass any authentication as can genuine tags [1]. If left unidentified, cloned tags pose a substantial threat to RFID applications that use the genuineness of tags to validate the authenticity of tagged objects. For example, carrying cloned tags, products in an RFID-enabled supply chain lead to financial losses [2], healthcare facilities in RFID-aided hospitals jeopardize personal safety [3], while RFID-incorporated passport cards even threaten national security [4]. However, existing prevention techniques based on cryptography and encryption (e.g., proposals in [5], [6], [7], [8]) are not affordable to most off-the-shelf low cost tags [9].

To secure RFID applications, RFID systems are soliciting solutions that can expose unauthentic objects by identifying their attached cloned tags. Although researchers have dedicated active efforts to RFID security and privacy and contributed exciting advances [1], designing fast cloned-tag identification protocols for applications in large-scale RFID systems is yet not thoroughly investigated. In this paper, we concentrate on the application scenario where tagged objects are confined in the same RFID system [10]. Such applications include, for example, people tracking in buildings with RFID-based entrance control systems [11] and healthcare facilities monitoring in RFID-aided hospitals [3].

978-1-4673-1298-1/12/$31.00 ©2012 IEEE

SYNChronized secret (*SYNC*) [10] is, to our knowledge, the only study on cloned-tag identification for applications that confine tagged objects in the same RFID system. SYNC maintains a map of tag IDs and corresponding random numbers. A reader writes a random number to a tag's memory each time it scans the tag and updates the map accordingly. The reader identifies a cloned ID if it scans a tag with the ID but with a different random number from the one in the map. However, collecting IDs (as well as random numbers) from all tags in a large-scale RFID system is very time-consuming [12], [13], [14]. High time efficiency is a long-standing goal for scalable protocols in favor of the explosion of RFID applications [15]. Furthermore, another concern is that transmitting tag IDs in the air may leak identity information, which should be protected in some privacy-sensitive RFID applications [9].

Inspired by SYNC, we seek to design time-efficient cloned-tag identification protocols for securing applications that confine tagged objects in large-scale RFID systems, catering for the explosion of RFID applications [15]. We summarize our approach and highlight its contributions to fast cloned-tag identification as follows.

1. Identify all cloned tags rather than simply detect some of them. We can thus secure applications that confine all tagged objects in the same RFID system [11], [3], [10]. As to applications that distribute tagged objects across multiple places [2], if we could locate the source where tagged objects are from, we can also leverage our approach to reject objects attached with cloned tags before they are distributed. (Cloned-tag identification protocols dedicated to such applications can be found in [16], [17], [18], [19].)

2. Leverage broadcast and collisions to identify cloned tags. The idea is intuitive but efficient—when we specify a tag with a certain ID to send a response, there exists its cloned peer(s) if a collision of multiple responses occurs. This idea relieves us from resorting to complex cryptography techniques.

3. Strive for time efficiency gains in the protocol design. We derive a time lower bound on cloned-tag identification and propose a series of protocols toward approaching it. Through eliminating ID broadcast and bypassing useless time slots, we propose ES-BID, a protocol with execution time of only 1.4 times the value of the time lower bound. Simulation results show that, compared with SYNC, ES-BID reduces the execution time by nearly an order of magnitude.

II. Preliminaries

A. Problem Statement

System model. Following SYNC in [10], we adopt a common RFID system model that fits RFID applications confining all tagged objects in the same system. The system consists of a backend server, some reader(s), and many tags. Tags are attached to objects; the genuineness of tags are used to validate the authenticity of tagged objects. Without tag cloning attacks, each tag has a unique ID. Tag IDs are stored on the server; readers communicate with the server via a secure link and have granted access to tag IDs [10]. When multiple readers are synchronized and scheduled, we can logically treat them as one [20]. We assume that readers and tags communicate with a power level high enough to suppress the background noise. Error correction coding against channel errors is beyond the scope of this paper.

Attacker model. We adopt the attacker model as in literature [10], [16], [17], [18], [19]. The attacker launches a cloning attack and attaches cloned tags to unauthentic objects, posing threats to RFID applications. Threats of concern are, for example, jeopardizing personal safety in hospitals with RFID-tagged healthcare facilities [3] and causing financial losses in RFID-enabled supply chains [2]. We consider the well-studied scenario where cloned tags behave exactly the same as genuine tags [1], [10], [16], [17], [18], [19].

Problem formulation. We formulate the cloned-tag identification problem as identifying all the IDs of cloned tags. As a cloned tag copies all data carried by a compromised tag [1], the compromised tag can pass any authentication and so can the cloned tag. So we turn to protocols that verify whether an ID corresponds to multiple tags. If an ID associates with more than one tag, the ID resides in a genuine tag and some cloned tag(s). In what follows, we call IDs of cloned tags *cloned IDs* and use it interchangeably with *cloned tags* wherever no ambiguity arises. Our goal is to design fast cloned-tag identification protocols for large-scale RFID systems.

B. Methodology Overview

We leverage broadcast and collisions to identify cloned tags. The idea is intuitive: When the reader broadcasts a query message that specifies only one tag to send a response, we can ensure that cloned tags exist if the reader receives a collision of multiple responses. Let t_c denote the transmission time of a response long enough to verify a collision. We derive T_lower, a time lower bound on cloned-tag identification protocols for verifying n tag IDs, as

$$T_\text{lower} = nt_\text{c}.$$

The time lower bound T_lower holds because by leveraging broadcast we require only the response state (i.e., collision or not) corresponding to each ID. In the Philips I-CODE system [21], a 10-bit string with error-detection (e.g., CRC) embedded is enough to verify a collision [12], which is much shorter than a 32-bit random number used by SYNC.

BID: A baseline protocol. The most intuitive way of specifying a tag to respond is simply broadcasting its ID in a query message. This idea forms the basis for the Broadcast-friendly cloned-tag IDentification protocol (*BID*), which we name it so because only the specified tag can "bid" for the query message through sending a response. In the BID design, the reader broadcasts one tag ID after another with each in a query message. Upon receiving the query message, a tag sends a response to the reader if its ID is identical with the queried one. If no cloned tag with the queried ID exists, the reader receives only one response; otherwise, the reader detects a collision and thus identifies the cloned ID. It takes the reader $t_\text{id}+t_\text{c}$ time to verify an ID, where t_id denotes the transmission time of a tag ID. We therefore derive T_B, the execution time of BID, as

$$T_\text{B} = (t_\text{id} + t_\text{c})n.$$

III. Fast Cloned-Tag Identification Protocols

In this section, we propose fast cloned-tag identification protocols that gain time efficiency through eliminating ID broadcast and bypassing useless time slots.

A. S-BID: Slotted BID

Motivation. Observing the execution time of BID and the time lower bound, we find that the gap between them (i.e., $T_\text{B} - T_\text{lower} = nt_\text{id}$) is totally due to the time for broadcasting tag IDs. It stands to reason that we seek to eliminate ID broadcast in order to reduce the execution time. Another motivation to eliminate ID broadcast is preserving identity privacy in favor of privacy-sensitive applications [9].

S-BID design. We adopt using *slotted Aloha* [22] to specify tags to respond without broadcasting their IDs. In slotted Aloha, the reader sends a query message containing the number f of time slots and a random seed r. Upon receiving the query message, each tag picks up a time slot with index $h(ID, r)\text{mod}f$ to send a response, where h is a hash function implemented on off-the-shelf tags. A time slot chosen by no tag, only one tag, or multiple tags is usually called an *empty slot*, a *singleton slot*, or a *collision slot*, respectively [12]. We refer to *empty*, *singleton*, and *collision* as *slot states*. Since readers have granted access to tag IDs stored on the server in the cloned-tag identification problem, we can determine exactly which IDs correspond to tag responses at a certain time. Then we can identify cloned tags if a collision occurs when only one tag response is expected according to slotted Aloha, without broadcasting tag IDs.

S-BID is an iterative protocol with each round containing two phases, *vector formation* and *vector matching*. In the vector formation phase, we form an *expected slot state vector* (denoted as V_e) using the slot states that we expect, and form a *received slot state vector* (denoted as V_r) using the slot states that we actually receive. We form V_e and V_r as follows.

$$V_\text{e}[i] = \begin{cases} 0, & \text{if } |\{ID \mid h(ID, r)\text{mod}f = i\}| \not= 1, \\ 1, & \text{if } |\{ID \mid h(ID, r)\text{mod}f = i\}| = 1, \end{cases}$$

where ID is in the set of to-be-verified IDs and $i \in [0, f-1]$.

$$V_\text{r}[i] = \begin{cases} 0, & \text{if slot } i \text{ is not a collision slot,} \\ 1, & \text{if slot } i \text{ is a collision slot.} \end{cases}$$

In the vector matching phase, S-BID identifies cloned IDs through capturing the slot state transitions from expected singleton slots to collision slots. Specifically, the reader compares the vectors V_{e} and V_{r} bit-wisely and identifies a cloned ID with $h(ID, r)\text{mod}f = i$ if $V_{\text{e}}[i] = V_{\text{r}}[i] = 1$. For saving time, we require tags with verified IDs to keep silent in further iterations using a 1-bit indicator at the end of each time slot. Each slot thus takes $t_{\text{c}} + t_{\text{s}}$ time, where t_{s} denotes the transmission time of a single bit. At the end of each iteration, S-BID deletes verified IDs from the set of to-be-verified IDs. S-BID terminates after it verifies all n tag IDs.

Execution time of S-BID. In each round, S-BID verifies IDs hashed into only expected singleton slots. In a query frame, the ratio of expected singleton slots is equal to that of singleton slots when there are no cloned tags. The optimal ratio of singleton slots in a query frame is $\frac{1}{e}$ when f is set to the number of to-be-read tags, where e is the natural constant [22]. The number of tags is equal to the number of IDs when no cloned tag exists. Because in each iteration we are aware of the number of to-be-verified IDs, it is feasible for us to satisfy the condition for the optimal ratio of expected singleton slots. Optimally, S-BID can verify $\frac{1}{e}$ of to-be-verified IDs in each iteration. We therefore derive the number n_j of to-be-verified IDs and the number f_j of time slots in the jth ($j \geq 1$) iteration as follows:

$$f_j = n_j \approx (1 - \frac{1}{e})^{j-1} n.$$

Using the knowledge that n_j is a positive integer, we derive the upper bound of iteration times, denoted as j_{max}, as follows:

$$n_j \geq 1 \Rightarrow (1 - \frac{1}{e})^{j-1} n \geq 1 \Rightarrow j \leq \frac{\ln n}{1 - \ln(e-1)} + 1 = j_{\text{max}}.$$

The jth iteration takes time for transmitting a query message and for f_j time slots. The query message comprises f_j and r_j, where r_j denotes the random seed r used in the jth iteration. Without loss of generality, we assume r_j is m bits long. Thus, the time for transmitting the query message is $(\log f_j + m)t_{\text{s}}$. Combined with $(t_{\text{c}} + t_{\text{s}})f_j$ taken by f_j time slots, the execution time of the jth iteration, denoted as $T_{\text{SB}j}$, is

$$T_{\text{SB}j} = f_j t_{\text{c}} + (f_j + \log f_j + m)t_{\text{s}}.$$

Combining the execution time of j_{max} iterations, we define the optimal execution time of S-BID, $T_{\text{SB}} = \sum_{j=1}^{j_{\text{max}}} T_{\text{SB}j}$, as

$$T_{\text{SB}} \approx (en - e + 1)t_{\text{c}} + (en - e + 1 + (\frac{\log n}{2} + m)j_{\text{max}})t_{\text{s}}.$$

B. ES-BID: Enhanced S-BID

Motivation. In each iteration, S-BID verifies IDs corresponding to only expected singleton slots (i.e., slots with $V_{\text{e}}[i] = 1$). The ratio of expected singleton slots in a query frame is only up to $\frac{1}{e} \approx 36.8\%$. Even more than $(1 - 36.8\%) = 63.2\%$ of slots therefore cannot benefit S-BID, wasting such a lot of time. The problem now becomes how to bypass those useless time slots. Intuitively, we can solve this problem by informing tags when to send responses. It faces two challenges to implement the idea. First, we need to inform which tags to respond and which tags not to. According to the S-BID design, only tag responses in expected singleton slots are useful. Therefore, tags choosing expected single slots should respond and other tags should not, making slots other than expected singleton ones empty. The second challenge is therefore to bypass those empty slots. Next we will present ES-BID to tackle the preceding challenges.

ES-BID design. To bypass useless time slots, ES-BID introduces two lightweight modifications to S-BID in each iteration. *First*, the reader broadcasts the vector V_{e} in addition to f_j and r_j in a query message. Upon receiving the query message, a tag decides to normally respond in the time slot with index $i = h(ID, r)\text{mod}f$ if $V_{\text{e}}[i] = 1$ and decides to keep silent otherwise. We by the first modification make slots with $V_{\text{e}}[i] = 0$ empty. *Second*, if a tag decides to respond, it calculates the slot index by $i' = \sum_{k=0}^{i} V_{\text{e}}[k] - 1$. The intuition is that, by looking up the vector V_{e}, a tag is aware of how many slots are taken by other tags before it responds and thus bypasses those empty slots through the above slot index recalculation. The tag then sends a response in slot i'. It takes t_{c} time to transmit a response long enough to verify a collision. After a tag responds, it keeps silent in further iterations because ES-BID can verify its ID. We therefore do not need a 1-bit indicator as we use in the S-BID design.

Receiving tag responses, the reader forms a received slot state vector, denoted as V_{r}', as follows:

$$V_{\text{r}}'[i] = \begin{cases} 0, & \text{if slot } i \text{ is a singleton slot,} \\ 1, & \text{if slot } i \text{ is a collision slot.} \end{cases}$$

The reader then checks the vector V_{r}' bit-wisely and identifies a cloned ID with $\sum_{k=0}^{h(ID,r)\text{mod}f} V_{\text{e}}[k] - 1 = i$ if $V_{\text{r}}'[i] = 1$.

Execution time of ES-BID. The execution time of ES-BID consists of the time for transmitting query messages in all iterations and for receiving tag responses in all time slots. ES-BID takes totally n time slots with each picked up by one of n IDs. As each time slot takes t_{c} time, n time slots take nt_{c} time. We also need to estimate the time for transmitting query messages in all iterations. Compared with S-BID that transmits only f_j and r_j, ES-BID transmits f_j bits more to broadcast the vector V_{e} in each query message. For j_{max} iterations, the time for transmitting query messages is

$$\sum_{j=1}^{j_{\text{max}}} (f_j + \log f_j + m)t_{\text{s}} \approx (en - e + 1 + (\frac{\log n}{2} + m)j_{\text{max}})t_{\text{s}}.$$

Combining the time for transmitting query messages and for receiving tag responses (i.e., nt_{c}), we therefore derive the optimal execution time of ES-BID as

$$T_{\text{E}} \approx nt_{\text{c}} + (en - e + 1 + (\frac{\log n}{2} + m)j_{\text{max}})t_{\text{s}}.$$

IV. Performance Evaluation

Simulation environment. According to the RFID system model in Section II-A, we configure the simulation environment as follows. A reader communicates with many tags, using a power level high enough to suppress the background noise.

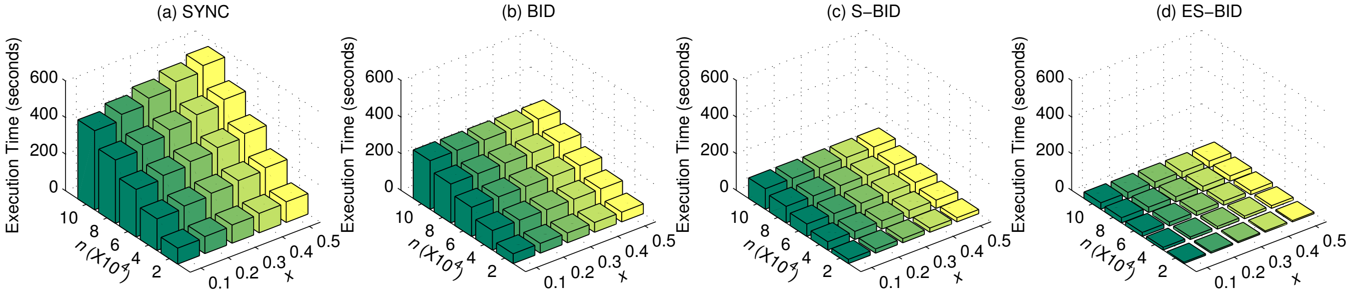

Fig. 1. Execution time comparison of SYNC, BID, S-BID, and ES-BID with varying number of tag IDs n and varying compromised tag ratio x.

A tag ID is 96-bit long [23]. The number n of tag IDs varies from 10,000 to 100,000. We randomly choose n integers from $[0,\ 2^{96}-1]$ as the tag IDs. Under a certain compromised tag ratio x, we randomly choose xn IDs as the cloned IDs. Then we run cloned-tag identification protocols to verify the tag IDs and to identify the cloned IDs. We use 32-bit random numbers for SYNC as the authors do in experiments [10]. We use 10-bit responses for our protocols to verify collisions [21]. The transmission time of a single bit is set to $t_{\text{s}} = 25$ µs [21]. We average the results over 100 trials.

Execution time. Figure 1 reports the execution time of SYNC, BID, S-BID, and ES-BID with varying number n of tag IDs and varying compromised tag ratio x. We can easily capture three obvious trends in Figure 1. First, if we prioritize the protocols in descending order of the time efficiency, the sequence is ES-BID > S-BID > BID > SYNC. On average, the execution time of ES-BID T_{ES} is only 1.4 times the time lower bound T_{lower}, being up to 91% less than the execution time of SYNC T_{S}. Second, the execution time of each protocol increases with n. Third, different from SYNC, our protocols (Figure 1(b)-(d)) are insensitive to the compromised tag ratio x. The insensitivity to the compromise tag ratio, which is hard to predict, is important to estimate a protocol's execution time and thus to envision whether the protocol is time-efficient.

V. Conclusion

We have studied the cloned-tag identification problem that is of practical importance to secure RFID applications. We concentrate on the application scenario where all tagged objects are confined in the same RFID system. To meet the time efficiency requirement for large-scale RFID systems, we seek to design protocols that can identify cloned tags as fast as possible. We leverage broadcast and collisions to identify the cloned tags. This approach gets rid of more complex cryptography techniques and time-consuming transmission of tag IDs, and hence is affordable to resource-constrained low cost tags. Based on this approach, we derive a time lower bound on cloned-tag identification protocols and strive for time efficiency gains when we design our protocols, toward approaching the time lower bound. Both analysis and simulation results show that all our protocols are faster than the existing protocol. Another merit of the proposed protocols is that they can benefit also applications that distribute tagged objects across multiple places.

References

[1] A. Juels, "RFID security and privacy: A research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[2] F. Kerschbaum and A. Sorniotti, "RFID-based supply chain partner authentication and key agreement," in *ACM WiSec*, 2009, pp. 41–50.

[3] B. Janz, M. Pitts, and R. 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, pp. 132–148, 2005.

[4] K. Koscher, A. Juels, V. Brajkovic, and T. Kohno, "EPC RFID tag security weaknesses and defenses: passport cards, enhanced drivers licenses, and beyond," in *ACM CCS*, 2009, pp. 33–42.

[5] L. Bolotnyy and G. Robins, "Physically unclonable function-based security and privacy in RFID systems," in *IEEE PerCom*, 2007.

[6] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications," in *IEEE RFID*, 2008.

[7] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *IEEE SecureComm*, 2006, pp. 59–66.

[8] A. Juels, "Strengthening EPC tags against cloning," in *ACM WiSe*, 2005, pp. 67–76.

[9] S. Spiekermann and S. Evdokimov, "Privacy enhancing technologies for RFID - A critical investigation of state of the art research," in *IEEE Privacy and Security*, 2009.

[10] M. Lehtonen, D. Ostojic, A. Ilic, and F. Michahelles, "Securing RFID systems by detecting tag cloning," *Pervasive Computing*, vol. 5538, pp. 291–308, 2009.

[11] K. Finkenzeller *et al.*, *RFID Handbook: Fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. Wiley, 2010.

[12] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *ACM MobiCom*, 2006, pp. 322–333.

[13] K. Bu, B. Xiao, Q. Xiao, and S. Chen, "Efficient pinpointing of misplaced tags in large RFID systems," in *IEEE SECON*, 2011, pp. 260–268.

[14] S. Chen, M. Zhang, and B. Xiao, "Efficient information collection protocols for sensor-augmented RFID networks," in *IEEE INFOCOM*, 2011, pp. 3101–3109.

[15] Explosive growth projected in next five years for RFID tags, http://www.instat.com/press.asp?ID=1545.

[16] R. Koh, E. Schuster, I. Chackrabarti, and A. Bellman, "Securing the pharmaceutical supply chain," *Auto-ID Center MIT, White Paper*, 2003.

[17] M. Lehtonen, F. Michahelles, and E. Fleisch, "How to detect cloned tags in a reliable way from incomplete RFID traces," in *IEEE RFID*, 2009, pp. 257–264.

[18] 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.

[19] D. Zanetti, L. Fellmann, and S. Capkun, "Privacy-preserving clone detection for RFID-enabled supply chains," in *IEEE RFID*, 2010.

[20] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *ACM MobiHoc*, 2010, pp. 1–10.

[21] Philips ICode system design guide, http://www.nxp.com/.

[22] L. Roberts, "ALOHA packet system with and without slots and capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[23] EPC class-1 generation-2 RFID protocol V.1.0.9, http://www.epcglobalinc.org/home.