Eingerprint: Robust Energy-related Fingerprinting for Passive RFID Tags

Xingyu Chen, Jia Liu, Xia Wang, Haisong Liu, Dong Jiang,
and Lijun Chen, *Nanjing University*

https://www.usenix.org/conference/nsdi20/presentation/chen

This paper is included in the Proceedings of the
17th USENIX Symposium on Networked Systems Design
and Implementation (NSDI '20)
February 25–27, 2020 • Santa Clara, CA, USA

978-1-939133-13-7

Eingerprint: Robust Energy-related Fingerprinting for Passive RFID Tags

Xingyu Chen* Jia Liu* Xia Wang Haisong Liu Dong Jiang Lijun Chen
State Key Laboratory for Novel Software Technology, Nanjing University, China

Abstract

RFID tag authentication is challenging because most advanced cryptographic algorithms cannot be afforded by passive tags. Recent physical-layer identification utilizes unique features of RF signals as the fingerprint to authenticate a tag. This approach is effective but difficult for practical use because it either requires a purpose-built device to extract the signal features or is sensitive to environmental conditions. In this paper, we present a new energy-related fingerprint called *Eingerprint* to authenticate passive tags with commodity RFID devices. The competitive advantage of Eingerprint is that it is fully compatible with the RFID standard EPC-global Gen2, which makes it more applicable and scalable in practice. Besides, it takes the electrical energy stored in a tag's resistor-capacitor (RC) circuit as the fingerprint, which is robust to environmental changes such as tag position, communication distance, transmit power, and multi-path effects. We propose a new metric called persistence time to indirectly estimate the energy level in the RC circuit. A select-query based scheme is designed to extract the persistence time by flipping and observing a flag in the tag's volatile memory. We implement a prototype of Eingerprint with commodity RFID devices without any modifications to the hardware or the firmware. Experiment results show that Eingerprint is able to achieve a high authentication accuracy of 99.4% when three persistence times are used, regardless of device diversity and environmental conditions.

1 Introduction

Radio frequency identification (RFID) is gaining increasing popularity in a wide range of applications, including warehouse inventory [15–17, 34], object tracking [13, 24, 25, 27, 30], and supply chain [22], due to its compelling features, dropping costs, and standardizations. Each RFID tag has a unique digital identity to label tagged items, brings item intelligence to our daily life, and allows the reach of the Internet to include objects as diverse as retail products, library books, debit cards, passports, driver licenses, car plates, and medical devices. In general, the RFID tags fall into two categories: active and passive. Active tags have their own power source and remain active all the time. Compared with the passive tags, they have more computational capabilities and longer read ranges. However, the built-in power source makes them bulky and expensive, which restricts these tags to high-end applications. In contrast, passive tags do not have a built-in power source and are powered by either induction or electromagnetic RF signals of the reader. They have limited computational capabilities and a lower read range than active tags. In spite of these limitations, they are common due to their low cost, small size, and longer life.

In recent years, with the proliferation of RFID systems, the problem of RFID security has attracted increasing attention. A great number of authentication protocols have been proposed to identify the authenticity of a tag [5, 8, 9, 19]. In the nascent stage, the authentication protocols check only the data (e.g., TID [4]) stored in a tag's memory, which is vulnerable to counterfeiting attacks: Adversaries can easily retrieve the data from a genuine tag with a commercial reader and forge a replica by filling its memory with the same data as the genuine tag. To address this problem, some cryptographic approaches are studied. By transmitting the ciphertext rather than the plaintext, the communication channel between a reader and a tag is protected against eavesdropping. However, this approach requires extra hardware components to support high computation overhead, which greatly increases the cost of a passive tag as well as reduces the communication range between the reader and the tag. Hence, it is rarely used by most commercial passive tags.

Motivated by the above limitations, recent research has shifted to physical-layer identification (PLI), which is commonly referred to as RF fingerprinting [10, 11, 20, 33, 35, 36]. It is the process of identifying a device based on transmission imperfections exhibited by its radio transceiver. The key appeal of applying RF fingerprint for authentication is twofold. First, RF fingerprints are unique and unpredictable, such that

[*] These authors contributed equally to this work

they can provide high security guarantees against various protocol-layer attacks. Second, no upgrades of hardware or firmware on existing systems are required, which makes it scalable to the wide use of RFID systems. In spite of this advancement, however, PLI suffers from two problems. First, most PLI-based solutions need a specialized device to detect physical-layer signals, which cannot be deployed in commodity RFIDs. Second, some work is not resistant to environmental or signal acquisition factors, e.g., RF phase values, a widely used metric for RF fingerprinting, heavily relies on the RF channels. Two different measurements of the same tag are very likely to give rise to different RF phases.

In this paper, we explore a brand-new fingerprint called energy-related fingerprint (*Eingerprint*) to authenticate passive tags with commodity RFID systems. Eingerprint takes the electrical energy stored in the tag's circuit as the fingerprint, which is robust to environmental conditions, including tag position, tag orientation, communication distance, transmit power, and multi-path effects. The basic idea is that passive tags do not have any built-in power source and are energized by the electromagnetic RF signals issued by the reader. To ensure proper functioning, a tag needs to store some electrical energy into its microchip, which is equivalent to a resistor-capacitor (RC) charging circuit [38]. Due to manufacturing imperfection, no two tags could ever have exactly the same RC circuit. If we can detect this difference, then we are able to fingerprint each tag as desired.

However, this is not easy. Building the electronic test circuit to physically measure the RC circuit of each tag is infeasible because it destroys the tag's structure and functions. Instead, we use a new metric called *persistence time* to indirectly reflect the RC circuit. The persistence time is the time span from the initial supply voltage when the RC circuit is fully charged decaying to a very low level that cannot afford the tag to run properly, which heavily relies on the RC circuit itself. In other words, if two RC circuits differ from each other, their persistence time is very likely to be different. On the basis of this idea, we design a Gen2-compatible approach to measure the persistence time based on a one-bit inventoried flag of a tag (a one-bit register in a tag's volatile memory). The volatile memory requires power to maintain the stored information. Once the power is cut off (or is lower than a threshold), the stored data are quickly lost. By flipping the inventoried flag and continuously observing its status with Gen-2 compatible commands, we are able to extract the persistence time of a tag. Afterwards, a t-test based model is designed to validate the genuineness of the tag. In addition, instead of individual fingerprinting, we propose a quick and reliable scheme to deal with multiple tags in parallel, which greatly improves the time efficiency of tag authentication. The main contributions of this paper are threefold.

- We explore a new energy-related fingerprint called *Eingerprint* to authenticate passive tags with commodity RFID devices. The competitive advantage of Eingerprint is that it is fully compatible with the RFID standard, which makes it more applicable and scalable for practical use. Besides, it takes the electrical energy as the fingerprint, which is robust to various environmental conditions.

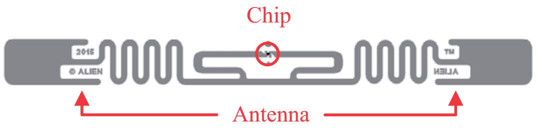


Figure 1: Alien Squiggle general-purpose RFID tag.

- We propose a new metric *persistence time* to indirectly indicate the energy level stored in a tag's RC circuit. A select-query based scheme is designed to measure the persistence time by flipping and observing a flag in the tag's volatile memory.
- We implement a prototype of Eingerprint in a commercial off-the-shelf RFID system with over 1000 tags. Extensive experiments show that our fingerprinting system is able to achieve a high accuracy of 97.3% and 99.4% when one persistence time and three persistence times are used respectively, without any changes to the hardware.

The rest of the paper is organized as follows. Section 2 overviews the fingerprinting model and proposes an energy-related fingerprint. Section 3 proposes a Gen2-compatible scheme to derive the fingerprint. Section 4 uses the fingerprint distribution to validate the genuineness of a tag. Section 5 evaluates the performance of the fingerprinting system. Section 6 introduces the related work. Finally, Section 7 concludes this work.

2 Overview

2.1 Fingerprinting Model

Passive tags do not have any built-in power source and are energized by the electromagnetic RF signals emitted by the reader. In general, a passive RFID tag consists of two components: the tag antenna and the microchip. As shown in Fig. 1, the microchip is usually placed right at the terminals of the tag antenna. When the RF signals are received by the tag antenna, the voltage developed on antenna terminals powers up the chip for computing and modulating the backscattered signal. The passive tag can be equivalent to a resistor-capacitor (RC) series circuit [38] that is composed of a resistor and a capacitor.

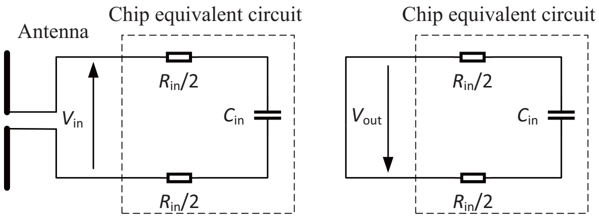
As a result of manufacturing imperfection, no two tags could ever have exactly the same microchip; the same idea applies to the electronic components (the resistor and the capacitor). If we can detect the difference of these electronic components among different tags, then we are able to fingerprint each tag from the physical-layer perspective, which forms the fingerprinting metric of this work. To achieve this goal, an intuitive solution is to set up an electronic test circuit and measure each electronic component manually. This concept works in theory but suffers from three problems in practice. First, a tag needs to be dissected (physically separating the microchip from the antenna), which damages the tag's structure and function. Second, performing measurements individually and manually is time consuming, especially when many tags need to be authenticated. Third, a purpose-built electronic test platform is needed to measure the chip circuit, which increases the cost of fingerprinting and is not scalable in commercial use. Hence, a new fingerprint that is able to reflect the attributes of the electronic components is required.

2.2 Fingerprint: Persistence Time

Consider the RC circuit of the microchip. When the tag captures the energy from the RF signals issued by the reader, it is actually an RC charging process. The equivalent charging circuit is shown in Fig. 2(a), where a capacitor C_{in} in series with a resistor R_{in} is connected across a DC battery supply (the power is obtained from the RF signals). The capacitor will gradually charge up through the resistor until the voltage across it reaches the supply voltage of the DC battery, namely, fully charged. According to the Gen2 standard, this charging process lasts for 2 ms at most. Afterwards, if we remove the voltage source (e.g., turn off the reader) from the fully charged circuit, the capacitor that is able to store the electrical energy acts like a small battery and releases the energy as required. This is referred to as an RC discharging process. As shown in Fig. 2(b), the capacitor discharges through the resistance in the opposite direction, which enables the tag to compute and communicate with the reader. As the discharge continues, the voltage goes down and there is less discharge current across the circuit. When the voltage decays to a very low level that cannot afford the tag to run properly, we say that the tag is exhausted and out of function. Assume that the initial supply voltage of the fully charged circuit is V_{in} and the minimal voltage that is needed to drive a tag is V_0. In the discharging stage, we refer to the time span from the initial supply voltage V_{in} decaying to the voltage threshold V_0 as *persistence time*, which can be derived as follows:

$$T_p = R_{in} \times C_{in} \times \ln(\frac{V_{in}}{V_{in} - V_0}), \tag{1}$$

where R_{in} and C_{in} are the resistance and capacitance of the microchip, respectively [38]. In Eq. (1), the V_0 voltage threshold is a constant when a tag chip is manufactured. For V_{in}, it varies with the available input power and thus depends on the energy captured by the tag antenna. This would be a variable in different communication conditions, e.g., different communication distances. To provide a stable voltage to the digital core, however, the commercial tag is required to carry a low dropout regulator [38], which uses a voltage reference block to produce a regulated and constant voltage V_{in}. Hence, the persistence time relies on the four constants R_{in}, C_{in}, V_0, and V_{in}, which are determined by the hardware of a tag, regardless of the environment factors, e.g., the communication distance, the tag location, multipath effects. By measuring the persistence time, we are able to figure out the difference of tag chips. This forms the basic idea of our method.



(a) Charging. (b) Discharging.

Figure 2: RC circuit of a tag's microchip.

The energy-based fingerprint has three competitive advantages. First, any Gen2-compatible readers are able to measure the persistence time of a commodity tag with no need for any modifications to the hardware or the firmware. Hence, implementing and deploying the fingerprinting system is easy in practice. Second, the persistence time not only accurately reflects the RC circuit of the tag chip but is also robust to the environment changes (e.g., the communication distance, the tag location, multipath effects), which is a key challenge for some PLI work [11, 28]). Third, a commercial tag has several independent persistence times (different RC circuits), which form different fingerprints to jointly authenticate the tag, thus making it hard to counterfeit. In spite of this advancement, measuring the persistence time of a tag is not easy. Next, we first show the system architecture of our approach and then detail how to obtain the persistence time in a Gen2-compatible commodity RFID system.

2.3 System Architecture

In general, the workflow of the fingerprinting system consists of three steps, which are shown in Fig. 3.

- *EPC Identification*: The reader interrogates a tag according to the Gen2 protocol and checks whether the EPC (i.e., tag ID) is identical to the tag to be authenticated. If no, then the tag is counterfeit. Otherwise, the system moves to the second step.
- *Fingerprint Extraction*: This step aims to extract the persistence time of the tag and treats it as the tag's energy fingerprint. Two key issues need to be solved. First, how can the persistence time be measured with the commercial RFID devices? Second, how can the time efficiency be improved and how can the measurement of multiple tags be conducted in parallel?
- *Genuineness Validation*: The system measures the persistence time and validates it with the stored records in the database. If it passes the authentication, then the tag is considered a genuine tag; otherwise, it is a counterfeit.

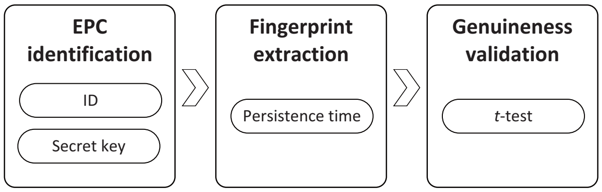


Figure 3: The workflow of the fingerprinting system.

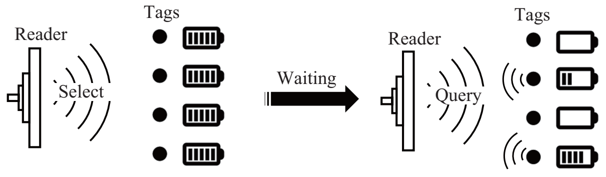


Figure 4: Basic idea of fingerprint extraction.

3 Fingerprint Extraction

3.1 Basic Idea

The basic idea of measuring the persistence time is to build a fully charged RC circuit and then run the discharging operation. This approach requires the following three steps. As shown in Fig. 4, first, we turn on the reader and let it issue the RF signals to energize the tag. Second, after the tag is fully charged, we cut off the power by turning off the reader; the discharging process starts. Third, after a period of time T_d, we check whether the tag is exhausted or not. By gradually increasing the time period T_d and repeating the above three steps, we can find a maximum of T_d that is guaranteed to help the tag work properly. This maximum is actually the persistence time to be measured.

Among above three steps, the first two, turning on and off the reader, are easy to operate. However, examining when the power of the tag is exhausted with a commodity RFID system is challenging. To address this problem, we resort to the volatile memory of a tag. Unlike non-volatile memory (e.g., NAND flash and solid-state drives), the volatile memory requires power to maintain the stored information. Once the power is cut off (or lower than a threshold), the stored data are quickly lost. In the RFID standard Gen2, we find a metric *inventoried flag*, which is a one-bit indicator in a tag's volatile memory. By flipping the inventoried flag and checking its status continuously, we are able to know when the power of the tag is exhausted. Next, we first introduce the Gen2 protocol and then detail how to measure the persistence time based on the RFID standard.

3.2 EPCglobal Gen2 Protocol

The EPCglobal Gen2 (Gen2) protocol is a worldwide UHF RFID standard that defines the physical interactions and logical operating procedures between the readers and tags [4]. On the basis of Gen2, we highlight the related functions that we will be involved by Eingerprint below.

Tag Memory. Gen2 standard specifies that the tag memory is supposed to contain four distinct memory banks (page 44—51 in [4]). MemBank-0 is reserved for kill and access passwords if encryption is implemented on the tag. MemBank-1 stores the electronic product code (EPC), i.e., tag ID that is often referred to. MemBank-2 stores TID that indicates the tag- and manufacturer-specific data at the time of manufacture, which is permalocked and unchangeable. MemBank-3 is user memory that allows customized data storage. In this work, we need to visit the tag's ID, so MemBank-1 is used.

Sessions & Inventoried Flags. Gen2 requires the readers and tags to provide four sessions (denoted as S0, S1, S2, and S3). Tags in one of these sessions shall neither use nor modify an inventoried flag for a different session. This allows two or more readers to use different sessions to independently inventory a common tag population (in different time slots). The inventoried flag is actually a one-bit indicator of a tag's volatile memory. The binary state of the inventoried flag is denoted by A and B, respectively, where A is the initial state as usual. The volatile memory requires power to maintain the stored information. Once the power is lower than a threshold, the stored data are quickly lost, that is, the inventoried flag will flip to A when the power of the tag is exhausted, no matter what the previous state is. According to Gen2, each session corresponds to an independent inventoried flag that needs different power levels to maintain its state, so the persistence time of each inventoried flag is different. Table 1 shows the persistence periods of different sessions specified by the Gen2 protocol. As we can see, when the tag is not energized, the persistence times of the inventoried flags in S2 and S3 are greater than 2 s. In contrast, the persistence time in S1 varies between 500 ms and 5 s, and no persistence time (always in A) is found for S0. This specification provides us with three persistence times by using the inventoried flags in S1, S2, and S3. Next, we take the inventoried flag in S1 as an example to show how our fingerprinting system works; the two other sessions can be used in the same way.

Select. *Select* is a mandatory command that is prior to each inventory round. It allows a reader to choose a specific subset of tags that participate in the subsequent inventoried round.

Table 1: Persistence time

G2 Session	Persistence time
S0	Tag energized: Indefinite Tag not energized: none
S1	Tag energized: 500 ms -5 sec Tag not energized: 500 ms -5 sec
S2	Tag energized: Indefinite Tag not energized: >2 sec
S3	Tag energized: Indefinite Tag not energized: >2 sec

Aside from tag selection, the *Select* command can also assert or deassert a tag's selected (SL) flag, or set a tag's inventoried flag to either A or B. These flags are used to determine whether or not a tag may respond to a reader. Specifically, a *Select* command consists six fields.

- *MemBank, Mask, Length, Pointer*. These four fields jointly determine which tags are matching or not. *MemBank* specifies which memory bank is chosen for comparison. As aforementioned, four memory banks are available, MemBank-0, MemBank-1, MemBank-2, and MemBank-3, which are indicated by 0, 1, 2, and 3, respectively. *Pointer* indicates the starting position in the chosen memory bank. *Length* determines the length of *Mask*, which is a customized bit string according to the user demands. If *Mask* is the same as the string that begins at *Pointer* and ends *Length* bits later in the memory of *MemBank*, then the corresponding tag is matched.
- *Target, Action.* The field *Target* indicates the object that *Select* will operate, which is either a tag's SL flag or an inventoried flag in any one of four sessions. The sessions are specified by the Gen2 protocol to fit the case of exclusive reading among multiple readers. Therefore, five different targets can be chosen. The selection function is actually achieved by masking the interested tags, setting the matching tags' inventoried flags or SL flag to a specific state while not-matching tags to opposite, and finally operating the tags with the same flag state. How to set the inventoried flag and the SL fag is determined by the *Action* field. As shown in Table 2, eight actions are available, where matching and not-matching tags set their inventoried flags to *A* or *B*. By combining *Target* and *Action*, the reader is able to modify the state of the inventoried flags or the SL flag for a group of tags. For example, when the *Action* is 0, the matching tags are set to *A* while the not-matching tags are set to B. The term "do nothing" means the tags keep their flags unchanged.

Query. *Query* command starts a new inventory round over the tag subpopulation, which are chosen by the previous *Select* command(s). In the inventory round, the reader will carry out a frame that consists of some time slots. Each "selected" tag randomly picks one of these time slots and transmits its tag ID to the reader in that slot. After a tag is queried by the reader, it will invert its inventoried flag, i.e., from the state *A* to *B*, or vice versa. *Query* includes three fields that we would like to focus on.

- *Session, Target*. Similar to that in *Select*, this field *Session* in *Query* specifies one of the four sessions used in the incoming inventory round. The field *Target* determines which tags will participate in the current inventory round, where 0 indicates the tags with the inventoried flag being *A* and 1 indicates *B*.

Table 2: Eight actions of *Select*.

Action	Tag Matching	Tag Not-Matching	Abbr.
000	assert **SL** or **inventoried** → *A*	deassert **SL** or **inventoried** → *B*	AB
001	assert **SL** or **inventoried** → *A*	do nothing	A-
010	do nothing	deassert **SL** or **inventoried** → *B*	-B
011	negate **SL** or *(A→B, B→A)*	do nothing	S-
100	deassert **SL** or **inventoried** → *B*	assert **SL** or **inventoried** → *A*	BA
101	deassert **SL** or **inventoried** → *B*	do nothing	B-
110	do nothing	assert **SL** or **inventoried** → *A*	-A
111	do nothing	negate **SL** or *(A→B, B→A)*	-S

- *Sel*. This field consists of two bits that determine which tags respond to *Query*: 00_2 and 01_2 indicate all matching tags in the previous Select command; 10_2 indicates tags with deasserted SL flag ($\sim SL$); and 11_2 indicates tags with asserted SL flag (SL).

On the basis of the above Gen2-compatible functions, we next detail how to jointly utilize the *Select* and *Query* commands to measure the persistence time by using the state of the inventoried flag. The method is called select-query based measurement.

3.3 Select-Query based Measurement (SQM)

The basic idea is that when the internal energy of a tag is exhausted, the inventoried flag will move back to the initial state *A* for sure, regardless of its previous state. If we set the tag's inventoried flag to *B* and keep the RC circuit fully charged, then the time period from starting discharging to the time when the inventoried flag turns to *A* can be treated as the persistence time.

3.3.1 Design of SQM

To measure the discharging time, we need to jointly use the *Select* command and the *Query* command. According to Gen2, a *Select* command can be written as follows:

$$S(\underbrace{t}_{\text{Target}}, \overbrace{a}^{\text{Action}}, \underbrace{b}_{\text{Membank}}, \overbrace{p}^{\text{Pointer}}, \underbrace{l}_{\text{Length}}, \overbrace{k}^{\text{Mask}}). \tag{2}$$

To set a tag's inventoried flag to *B*, the reader just needs to broadcast a *Select* as follows:

$$\text{Flag} \leftarrow \text{BA: } S(1,4,1,32,96,id), \tag{3}$$

where $t = 1$ (001_2) means the operating object is set to the inventoried flag in session 1 (S1), $a = 4$ indicates that the inventoried flags of matching tags will be set to *B*, while those of not-matching tags will be set to *A*, $(b,p,l,k) = (1,32,96,id)$ means the tag's ID is the same as *id* is selected (matching). Note that the first bit of the tag ID starts from the 32nd bit ($p = 32$) in MemBank-1, because the first 32 bits are a protocol-control (PC) word and the tag ID follows behind the PC word. More details can be seen in [4].

By this means, the target tag is set to *B*. Now the question is how long we can obtain a fully charged RC circuit. Gen2 specifies that the charging time should be no longer than 2 ms, which is much less than the time period (about 20 ms) for broadcasting a select command. In other words, once the select command in 3 is carried out, the target tag has the inventoried flag being B and also the RC circuit being fully charged.

Afterwards, we move to the discharging process by turning off the readers. Given that the tag cannot harvest energy from the reader anymore, the stored electric energy is consumed gradually. After a period of time T_d for discharging, the reader broadcasts a query command to check whether any tag with the inventoried flag B exists. The query command is as follows:

$$\text{Query } B: \ Q(Session = 1, Target = 1, Sel = 0). \quad (4)$$

If a tag reply is received, it means that the persistence time of this tag is longer than T_d. In this case, we need to increase T_d by a small step Δ_t and repeat the above select-query process again. For the first time period T_d that makes no tag reply, it is treated as the persistence time to be measured. That is because no tag reply means that the tag's inventoried flag has flipped to A since the power ran out. Note that, for the session 1 (S1), since the persistence time is bounded between 500 ms to 5 s, we can initialize T_d with 500 ms and increases it gradually until no tag reply occurs.

3.3.2 Multiple Tags

So far, we have discussed how to obtain the persistence time of a session for a single tag. In a practical scenario, however, authenticating multiple tags at a time is common. One intuitive solution is to fingerprint each tag in sequence, one by one. This works but suffers from high time latency. To make SQM more efficient and scalable to the multi-tag case, we need to deal with multiple tags in parallel.

An important observation is that broadcasting the select and executing the query operation take only a few milliseconds; the majority of the time overhead comes from trying the waiting period T_d. If we can let multiple tags wait concurrently, the execution time will decline sharply. Following this idea, we first set all target tags' inventoried flags to the state B, instead of individually dealing with one tag at a time. Afterwards, these target tags move to the discharging process and the energy is consumed gradually. After a period of time T_d, we query the tags with flag B as is. If a tag does not respond to the reader, its persistence time is T_d. This process repeats until all target tags are measured. In this way, the long discharging process executes in parallel, which saves a large amount of time overhead. For example, assume that we fingerprint 10 tags in parallel. We can reduce the waiting periods by about 90%; the global authentication performance is much better than the individual authentication performance.

Now the question is how to select a subset of tags and set their inventoried flags to the state B. Assume there are n tags, in which m tags are target tags. We can separate these m tags from the entire tag set via m select commands. The selection process is executed as follows. We first use the $Action = BA$ to select the first tag t_1, i.e., t_1's inventoried flag is set to B while others are A. Afterwards, for the ith tag t_i, the $Action$ is set to $B-$. We use $B-$ because this action will set the matching tag t_i to B accordingly but not change the settings of the previous tags. The commands are shown below.

$$\begin{array}{l} \textcircled{1}\ t_1 \leftarrow BA : S(2, a = 4, 1, 32, 96, id_1) \\ \textcircled{\scriptsize i}\ t_i \leftarrow B- : S(2, a = 5, 1, 32, 96, id_i), \ \ i \in [2, m], \end{array} \quad (5)$$

where id_i represents the tag t_i's tag ID, $a = 5$ means the action $B-$, which can be seen in Table 2. Besides, by investigating commodity RFID readers through their data sheets and real experiments, we find that these readers allow multiple selects to be broadcast in one transmission, e.g., two by Impinj R420 [2] and four by ALR 9900+ and ALR F800 [1]. With this function, we are able to fill several selects into a single one, further saving the communication overhead.

3.4 Enhanced SQM

Although concurrently fingerprinting multiple tags can sharply shorten the authentication time, a large gap still exists between SQM and efficient authentication primarily because that the process of increasingly adjusting the waiting time T_d is time-consuming. For example, assume a tag's persistence time is 3 s and the step length is 0.1 s. The waiting time T_d is initialized to 0.5 s and SQM needs to iteratively try 0.5 s, 0.6 s, 0.7 s, ..., 3.0 s. Summing up the overhead of each try, we have the overall time cost 45.5 s. This time cost is fine for some applications without real-time requirements. However, in the applications such as access control systems, this time is too long to be applicable for practical use.

The basic reason for the low time efficiency is that when a waiting time T_d is examined, we need to reset all tags and retry the next one. A longer time is needed for checking. If we can run the measurement within only one waiting time window, the performance will be improved greatly. Through extensive experiments, we find that the query command does not charge the tag in the session S1. In other words, during the discharging process, we are able to keep querying the tags, with no need to turn off the reader. Once a tag is queried by the reader, it will be recharged again.

3.4.1 Design of Enhanced SQM

With these features, the enhanced SQM measures the tag t_1's persistence time as follows. First, similar to the basic SQM, the reader broadcasts a select command (see Eq. (3)) with action BA to set t_1's inventoried flag to B. After that, the discharging process starts and the reader queries the tag with the flag state being A. The query command is

$$\text{Query } A: \ Q(Session = 1, Target = 0). \quad (6)$$

As shown in Fig. 5, during the discharging process, the internal circuit energizes the tag and keeps the inventoried flag

B, so the reader cannot receive any response from the tag t_1. When the power level is too low to maintain the information of the volatile memory, the inventoried flag moves back to the initial state A. At that time, because the reader keeps querying tags with A, the tag t_1 satisfying this condition will reply to the reader. By observing the time span from the start of discharging to the tag reply, we are able to derive the persistence time of the tag. Clearly, enhanced SQM does not need to try different waiting times; only one time window is able to measure the persistence time, which saves a great number of overheads. For example, consider the above tag with 3 s persistence time. Enhanced SQM results in great performance improvement, reducing the time from 44.5 s to only 3 s, in comparison to the basic SQM.

After responding to the reader, the tag flips its inventoried flag to B (according to Gen2); meanwhile, the RC circuit is fully charged. With the reader continuing to query A, the tag will reply after another persistence time. Hence, if we need multiple measures of persistence time, we just need to record each time interval between two adjacent tag responses, which is shown in Fig. 5. In fact, we can also simplify the enhanced SQM by removing the select command, that is, the reader directly enters the inventory stage. By keeping querying tags with A, the reader is able to get each tag's replies. The time interval between any two adjacent tag replies is the tag's persistence time. In addition, enhanced SQM can be extended to the multi-tag case, with no need for any modifications to the measurement process.

3.4.2 Multiple Tags

In spite of advancements, the enhanced SQM faces a new challenge in which the tags beyond the target tags might have negative effects on the measurement of the persistence time, especially when a great number of these tags exist. More specifically, assume that the tag set is τ and $\tau' \subseteq \tau$ is a subset of tags to be authenticated. The problem is that, when we set the inventoried flags of τ' to B with the select command, the tags in $\tau - \tau'$ will be set to A. In the follow-up inventory stage, the reader queries tags with flags being A; these tags $\tau - \tau'$ will attend to respond. As a result, the tags in τ' cannot give a prompt reply when their flags move back to A due to lack of energy. Setting $\tau - \tau'$ to B initially does not work either because these tags will still reply to the reader after their power level is lower than a threshold.

To address this problem, we resort to another indicator: SL flag. As mentioned previously, the SL flag has two states denoted by SL and $\sim SL$. The reader can specify a set of tags in one of the two states, which will participate in the inventory round. The SL flag and the inventoried flag are independent and can be jointly used to remove the interference of $\tau - \tau'$. The solution is to set the target tags τ' to SL while others $\tau - \tau'$ to $\sim SL$. In the inventory stage, we let only the tags with SL participate in the response. By this means, even if a tag in $\tau - \tau'$ is with the inventoried flag A, it has to keep silent to the command of querying A. Specifically, assume that $\tau' = \{t_1, t_2, ..., t_m\}$. The reader broadcasts the select commands as follows:

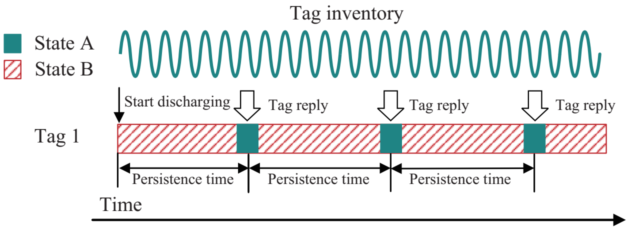


Figure 5: Enhanced SQM for obtaining the persistence time.

$$\begin{aligned} &\textcircled{1}\ t_1 \leftarrow AB : S(t = 4, 0, 1, 32, 96, id_1), \\ &\textcircled{i}\ t_i \leftarrow A- : S(t = 4, 1, 1, 32, 96, id_i),\ \ i \in [2, m], \end{aligned} \tag{7}$$

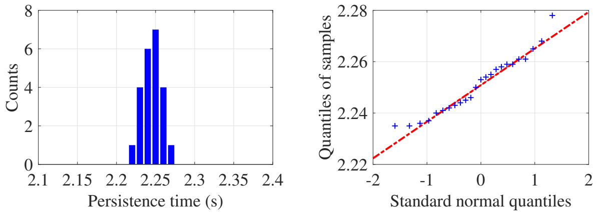
where $Target$ being set to 4 ($t = 4$) means that the operating object of the select is the SL flag. With the above select commands, the SL flags of the tags in τ' are asserted (SL), whereas those of tags in $\tau - \tau'$ are deasserted ($\sim SL$). Afterwards, we move to the inventory stage with the query command:

$$\text{Query } A \ \&\ SL: \ Q(1, Target = 0, Sel = 3), \tag{8}$$

where the fields $Sel = 3$ and $Target = 0$ mean that the reader queries only the tags with the inventoried flags being A together with asserted SL. In such a context, only the target tags of τ' have the chance to reply; other tags in $\tau - \tau'$ are silenced due to $\sim SL$. For any target tag, by recording the time interval between two adjacent replies, we can get its persistence time, which is treated as the energy-related fingerprint.

3.5 Degree of Parallelism

Simultaneous authentication of multiple tags greatly saves the time overhead. However, this is not free; it lowers the sampling rate of each tag when measuring its persistence time. That is because the read throughput of a reader model (how fast the reader can read the tags) is usually fixed. More tags correspond to reduced likelihood that a tag is read. A low sampling rate means a low resolution of measured persistence time, which further affects the authentication accuracy. To address this problem, we can partition a tag set into several small subsets if a large number of tags are to be authenticated. Afterwards, we deal with each subset of tags at a time. The process of fingerprinting a subset of tags can be seen in Section 3.3.2 (for SQM) and Section 3.4.2 (for enhanced SQM). This process repeats until all tags are validated. Note that the degree of parallelism is related to the read throughput of a reader. High read throughput ensures that more tags can be fingerprinted simultaneously. The degree of parallelism is evaluated in Section 5.3.2.



(a) Sample data.

(b) Q-Q plot.

Figure 6: Gaussian distribution of persistence time.

4 Genuineness Validation

To validate the genuineness of a tag, we need to compare its fingerprints under testing with those in the check-in stage. In this work, we take the distribution of the persistence time as the metric to perform the comparison. Intuitively, if a tag under testing is genuine, then its persistence time should follow the same distribution as its genuine records. Given a newly measured set X' of the persistence time and a genuine record X, the task of genuineness validation is reduced to verify whether X' and X follow the same distribution.

Now, we set up an RFID system that contains 1000 tags with eight different models supplied by three leading RFID companies: Alien [1], NXP [3], and Impinj [2]. For each tag, we run the enhanced SQM to obtain at least 20 measures of the persistence time. In Fig. 6(a), we randomly pick a tag and plot its persistence time. As we can see, the persistence time is likely to be a Gaussian distribution. We validate this conclusion through a quantile-quantile (Q-Q) plot, which is widely used to compare the similarity between two probability distributions. If two compared distributions are similar, then the points in the Q-Q plot will nearly lie on a line. As shown in Fig. 6(b), we compare the persistence time with the standard normal distribution. Clearly, the plots almost form a straight line, suggesting that the persistence time follows a Gaussian distribution.

In statistics, t-test is most commonly applied to determine whether the means of two sets of data with Gaussian distribution are significantly different from each other. Suppose the recorded data X follows a Gaussian distribution $N(\mu, \delta^2)$ and the data X' under testing follows a Gaussian distribution $N(\mu', \delta'^2)$. If the tag is a genuine tag, then μ' and δ'^2 are supposed to be very close to μ and δ^2, respectively. According to t-distribution, the mean value $\bar{X}'$ shall be

$$f(\bar{X}') = \frac{\bar{X}' - \mu}{\delta / \sqrt{n}}. \tag{9}$$

The t-test uses the significance level p as a threshold to determine whether or not accept $\bar{X}'$. The significance level p belongs to the interval [0, 1] and is typically set to 0.05 or less [23]. The setting of p will be discussed in Section 5.3.1.

Note that if the persistence time does not follow normal distribution, we can resort to a non-parametric test, e.g., Wilcoxon rank-sum test, which is valid for both non-normally distributed data and normally distributed data.

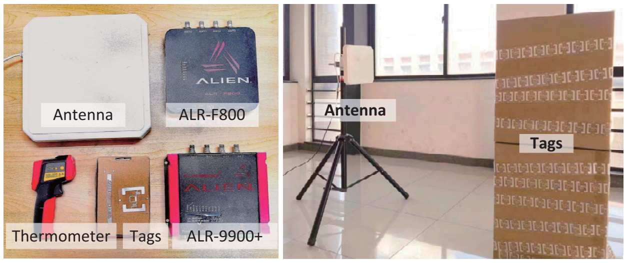


Figure 7: System deployment.

5 Implementation & Evaluation

In this section, we implement a prototype of Eingerprint in a commodity RFID system. On the basis of this system, we evaluate the performance of Eingerprint through extensive experiments in terms of the robustness to environmental changes and authentication accuracy.

5.1 System Deployment

The system setup is shown in Fig. 7. Two reader models, ALR-F800 and ALR-9900+ supplied by Alien [1], are employed in our experiment without any modifications to the hardware or the firmware. The reader is connected to a directional antenna (Laird S9028 [14], with a gain of 8.5 dBi) and operates at around 920 MHz. Over 1000 tags with 8 tag models are used in total. The model ALN-9634 [1] is adopted as the default in the experiments without explicit instructions. The development software of the fingerprinting system is Java, which adopts the Low-Level Reader Protocol (LLRP), specified by EPCglobal in its EPC Gen2 standard, to communicate. The host computer is a laptop with an Intel Core i5-8250U 1.8 GHz CPU and 8 GB RAM.

5.2 Impact of Environmental Factors

Resilience to environmental conditions is where Eingerprint shines, which is a basis for practical use. In this subsection, we investigate the impact of environmental factors on the measure of the persistence time, including the communication distance, tag orientation, communication frequency, transmit power, and temperature. All results are evaluated based on the inventoried flag in session 1 (S1) without explicit instructions. Similar conclusions can also be drawn in session 2 (S2) and session 3 (S3).

Distance. The communication distance between a reader and a tag is well known to have a great impact on the RF signals, e.g., RSSI or the phase value. To investigate the impact of the distance on Eingerprint, we vary the distance d and

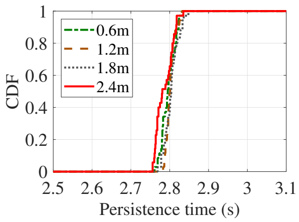


Figure 8: Impact of distance.

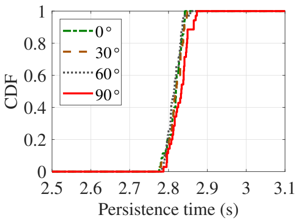


Figure 9: Impact of tag orientation.

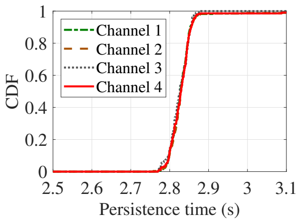


Figure 10: Impact of channel.

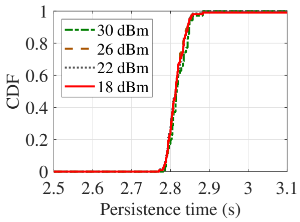


Figure 11: Impact of transmit power.

observe the changes in CDFs of persistence times. In this experiment, four distances are tested, where d_1 =0.6 m, d_2 =1.2 m, d_3 =1.8 m, and d_4 =2.4 m. As shown in Fig. 8, we can see that the CDFs are very close to each other and the means of the persistence times under the four distances are 2.797 s, 2.801 s, 2.804 s, and 2.787 s, respectively. These positive results demonstrate that the distance between a reader and a tag has little effect on the energy-related fingerprint.

Tag orientation. In some existing RF-based work [10,35], the authentication accuracy largely depends on the tag orientation. In Fig. 9, we observe the persistence time of a tag under different rotation angles, i.e., $0°, 30°, 60°, 90°$. The means of the measured persistence times are 2.817 s, 2.818 s, 2.814 s, and 2.814 s, respectively, corresponding to the four rotation angles. Similarly, the consistent results indicate that our energy-related fingerprint remains stable, regardless of the tag's rotation angles.

RF channels. A typical UHF reader has 16 channels working at 920—924 MHz ISM band. RF phase values, a widely used metric for RF fingerprinting, heavily rely on the RF channels. To examine whether the channel affects the stability of Eingerprint, we extract the persistence time from a tag under four different channels, where $channel_1 = 920.625$ MHz, $channel_2 = 921.625$ MHz, $channel_3 = 922.625$ MHz, $channel_4 = 923.625$ MHz. Fig. 10 shows the CDFs of the persistence times under the four channels. The close results demonstrate that the energy-related fingerprint is resistant to the communication channel.

Transmit power. Next, we examine the effect of the transmit power of the reader. In this experiment, we set the transmit power to 30 dBm, 26 dBm, 22 dBm, and 18 dBm respectively, and observe its impact on persistence time. As shown in Fig. 11, the CDFs of the persistence times under different transmit powers approach to each other. The positive results demonstrate that the transmit power has little impact on the energy-related fingerprint.

Temperature. In this experiment, we study the impact of the temperature on the persistence time. Four temperatures are investigated, three of which are close to each other (20 °C, 21 °C, 22 °C) and another is much higher (30 °C). Fig. 12 shows the CDFs of the persistence time under these four temperatures. As we can see, the three CDFs of temperatures 20 °C, 21 °C, 22 °C are similar, while that of 30 °C is different from the others'. This result indicates that the temperature has an impact on the persistence time. If the temperature change is slight, the impact could be negligible. Otherwise, we need to take the temperature into account if it varies considerably. This accords with the theory as temperature could affect the resistance and capacitance of electronic components. In fact, it is a blessing in disguise to some extent: each temperature corresponds to a fingerprint, which provides us with more fingerprints and higher authentication accuracy.

On top of the above experiments, we draw the conclusion that the energy-related fingerprint is resistant to various environmental conditions, including communication distance, tag orientation, communication frequency, and transmit power, except for the temperature.

5.3 Authentication Performance

In the experiments, three widely used metrics are applied to evaluate the authentication performance of Eingerprint, including false acceptance rate (FAR), false rejection rate (FRR), and authentication accuracy. FAR indicates the likelihood that the system will incorrectly accept a counterfeit. FRR indicates the likelihood that the system will fail to accept a genuine tag. For each experiment, we randomly pick two tags from 200 tags and treat one of them as a genuine tag and the other as a counterfeit. By checking whether each of them is genuine or not, we can record the number of correct checks. Repeating the above experiment 500 times, we derive the authentication accuracy that is equal to the ratio of the number of correct checks to the number of tests in total.

5.3.1 Significance Level

Eingerprint utilizes the significance level (threshold), denoted by p, to determine whether a testing fingerprint is valid or not. A large p is likely to reject a valid tag, leading to a high FRR, while a small p cannot figure out friend (genuine tag) or foe (counterfeit), increasing FAR. This dilemma requires a proper value of p to balance FRR and FAR. We extract fingerprints from 200 tags and respectively compute FRR and FAR under various p, which ranges from 0.01 to 0.06. As shown in Fig. 13, we set the value p to the value that corresponds to the intersect point of the two curves of FRR and FAR, i.e., $p = 0.03$, which is used in the following experiments.

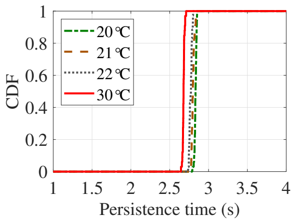


Figure 12: Impact of temperature.

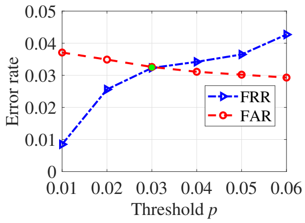


Figure 13: Threshold setting.

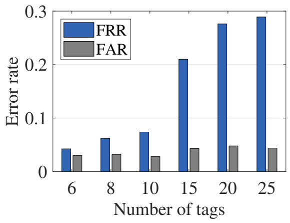


Figure 15: Multiple tags.

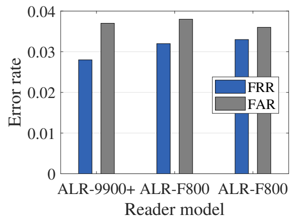


Figure 16: Device diversity.

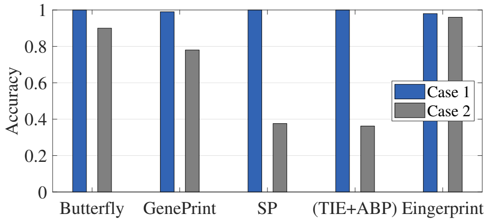


Figure 14: Performance comparison.

5.3.2 Authentication Accuracy

We now compare the authentication accuracy of Eingerprint with the state-of-the-art, including Butterfly [11], GenePrint [10], spectral feature (SP) [35], and time interval error together with the average base band power (TIE+ABP) [35]. Two cases are taken into account. In case 1, the tags are registered and authenticated at the same position. In case 2, the tags are registered and authenticated in different rooms. As shown in Fig. 14, all methods achieve a high authentication accuracy in case 1. However, the environmental changes in case 2 have a great impact on the performance of existing work. For example, the accuracy of SP drops sharply from 100% to 37.6%. By contrast, Eingerprint is resistant to these changes; the authentication accuracy in case 2 reaches 96.2%. Eingerprint is also more scalable than these approaches, which require a purpose-built device to measure the RF signals and cannot be deployed in a commercial RFID system. Notably, we just use one session to do the authentication. If more sessions are taken into account, the accuracy will be further improved, which will be shown next.

Selection of sessions. According to the Gen2 standard, three sessions with different persistence times can be used for tag authentication: session 1 (S1), session 2 (S2), and session 3 (S3). As shown in Table 3, we increasingly use these three sessions. The authentication accuracy improves as the number of sessions increases. This result is intuitive because more fingerprints reduce the probability that the system incorrectly accepts a counterfeit. Using multiple sessions, however, increases the authentication time. Hence, it is a trade-off between the accuracy and the time efficiency.

Table 3: Accuracy with different sessions

	S1	S1+S3	S1+S2+S3
Accuracy	97.3%	98.3%	99.4%

Multiple tags. We now study the performance of Eingerprint when authenticating multiple tags concurrently. We randomly choose 6, 8, 10, 15, 20, and 25 tags from 200 tags and authenticate them concurrently. As shown in Fig. 15, FRR sees a sharp rise as the number of tags increases because the large number lowers the sampling rate of each tag, which further lowers the resolution of the measured persistence time. In other words, the same tag is likely to have some persistence times apart from each other due to the low resolution, which increases the probability that a genuine tag is rejected. In contrast, the number of tags has a much lower impact on FAR because the same tag still more easily has similar persistence times than others even though the resolution is low. In addition, we can see that our method has potential in validating multiple tags in parallel. For example, when fingerprinting 10 tags, the FRR is 7.2%, the FAR is 2.3%, and the authentication accuracy is 95.2%. We assert that the degree of parallelism is related to how fast a reader can read tags. In this experiment, the read throughput of the reader is about 150 tags/s. If a faster reader is adopted, then the degree of parallelism could be higher.

Device diversity. In practice, using different devices to register and validate tags is common. To study the impact of device diversity, four readers are used, namely, three ALR-F800 readers and an ALR-9900+ reader. In the experiment, we first register 200 tags with an ALR-F800 reader and then validate the tags with the other three readers. As shown in Fig. 16, the authentication accuracy remains almost unchanged, regardless of which reader is used. This experimental result shows that device diversity has little impact on the performance of Eingerprint.

Tag model. We further study the performance of Eingerprint on different tag models. In the experiment, we test eight tag models from three leading RFID tag providers, which are Alien [1], NXP [3], and Impinj [2]. As shown in Table 4, Eingerprint achieves a high authentication accuracy (>94%) on all Alien and NXP tags. However, for Impinj tags, the accuracy experiences a sharp drop. By checking the persistence time, we find that the difference of persistence times of Impinj tags is much smaller than that of the other two. Hence, we recommend using Alien tags or NXP tags if tag authentication is required.

6 Related Work

Existing studies on RFID authentication can be divided into two categories: cryptographic-based approach [5, 8, 9, 18, 19, 32] and physical-layer identification (PLI) [6, 7, 10–12, 21, 26, 29, 31, 35–37, 40]. The former uses cryptographic technique to protect the communication between reader and tags against eavesdropping. However, existing cryptographic-based approaches suffer from two limitations. First, some cryptographic algorithms require high computation overhead, which is too heavy to be afforded by a passive tag [8]. Besides, it increases the cost of a passive tag as well as reduces the communication range between a reader and a tag. Second, some cryptographic-based methods are vulnerable to protocol-layer attacks, such as reverse engineering, side-channel, replay attack, and cloning [18, 32].

PLI is commonly referred to as RF fingerprinting, which utilizes the physical-layer information to identify digital devices. PLI has two advantages over cryptographic-based methods. First, the feature from the physical layer is unique and unpredictable, such that it can provide high security guarantees against various protocol-layer attacks. Second, no hardware or firmware upgrades on existing systems are required. Existing PLI work generally has three categories: location-based RF fingerprinting (LRF) [26, 31, 37], transient-based and preamble-based RF fingerprinting (TPF) [7, 10, 11, 29], and modulation error-based RF fingerprinting (MEF) [6, 12, 35].

LRF takes the location information as the fingerprint to authenticate a target, which works but strongly relies on the target's location. TPF fingerprints a device through the uniqueness of a certain fixed segment extracted from its transition signals and preamble signals [7, 10, 11, 29]. Since the transient-based and preamble-based features are always derived by spectral transformations, this approach is sensitive to environmental changes [29, 39]. MEF fingerprints a device through the modulation errors caused by hardware imperfection, such as SYNC correlation [6], carrier frequency offset [12], and time interval errors [35], which is channel-robust but usually requires a purpose-built device (e.g., USRP) to acquire fine-gain signal features and is thus not scalable to a commodity RFID system.

Table 4: Performance on different tag models

Company	Chip	Model	Accuracy
Alien	Higgs 3	ALN-9634	97.3%
	Higgs 4	ALN-9740	96.9%
	Higgs EC	ALN-9830	96.6%
NXP	Ucode G2iL	MiniWeb	94.4%
	Ucode G2iM	AD-380iM	94.9%
	Ucode 8	AD-238U8	94.2%
Impinj	Monza 4	H47	77.8%
	Monza R6	BLING	80.4%

7 Conclusion

In this paper, we propose a robust RFID authentication scheme by using an energy-related fingerprint. The basic idea is using the electric energy stored in a tag's circuit rather than RF signals as the fingerprint, which is resistant to the environmental changes. Directly measuring the tag's circuit to obtain the stored energy is impractical. Instead, we find an equivalent metric, namely, persistence time, that can reflect the circuit diversity indirectly. We design a Gen2-compatible select-query method to measure the persistence time. Afterwards, we use a t-test based model to validate the genuineness of a tag. We set up a prototype of the fingerprinting system, and extensive experiment results show that our system is able to achieve a high authentication accuracy of 99.4%, regardless of environmental conditions and without any hardware or firmware modifications.

Acknowledgments

We would like to thank our shepherd, Prof. Aaron Schulman, and the anonymous reviewers for their valuable feedback on an earlier draft of this paper. This work is supported by National Natural Science Foundation of China (Nos. 61702257 and 61771236), Natural Science Foundation of Jiangsu Province (BK20170648), Jiangsu Key R&D Plan (Industry Foresight and Common Key Technology, BE2017154), Fundamental Research Funds for the Central Universities, and Collaborative Innovation Center of Novel Software Technology and Industrialization. Jia Liu and Lijun Chen are the corresponding authors.

References

[1] Alien. `http://www.alientechnology.com`.

[2] Impinj. `http://www.impinj.com`.

[3] NXP. `https://www.nxp.com`.

[4] *GS1 EPCglobal. EPC radio-frequency identity protocols generation-2 UHF RFID version 2.0.1*, 2015.

[5] Karim Baghery, Behzad Abdolmaleki, Shahram Khazaei, and Mohammad Reza Aref. Breaking anonymity of some recent lightweight RFID authentication protocols. *Wireless Networks*, 25(3):1235–1252, 2019.

[6] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proc. of ACM MobiCom*, pages 116–127, 2008.

[7] Songlin Chen, Feiyi Xie, Yi Chen, Huanhuan Song, and Hong Wen. Identification of wireless transceiver devices using radio frequency RF fingerprinting based on

STFT analysis to enhance authentication security. In *Proc. of IEEE EMC+SIPI*, pages 1–5, 2017.

[8] Jung Sik Cho, Sang-Soo Yeo, and Sung Kwon Kim. Securing against brute-force attack: A hash-based RFID mutual authentication protocol using a secret value. *Computer Communications*, 34(3):391–397, 2011.

[9] Lijun Gao, Maode Ma, Yantai Shu, and Yuhua Wei. An ultralightweight RFID authentication protocol with CRC and permutation. *Journal of Network and Computer Applications*, 41:37–46, 2014.

[10] Jinsong Han, Chen Qian, Panlong Yang, Dan Ma, Zhiping Jiang, Wei Xi, and Jizhong Zhao. GenePrint: Generic and accurate physical-layer identification for UHF RFID tags. *IEEE/ACM Transactions on Networking*, 24(2):846–858, 2015.

[11] Jinsong Han, Chen Qian, Yuqin Yang, Ge Wang, Han Ding, Xin Li, and Kui Ren. Butterfly: Environment-Independent physical-layer authentication for passive RFID. In *Proc. of ACM UbiComp*, pages 1–21, 2018.

[12] Jingyu Hua, Hongyi Sun, Zhenyu Shen, Zhiyun Qian, and Sheng Zhong. Accurate and efficient wireless device fingerprinting using channel state information. In *Proc. of IEEE INFOCOM*, pages 1700–1708, 2018.

[13] Kiran Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti. WiDeo: Fine-grained device-free motion tracing using RF backscatter. In *Proc. of USENIX NSDI*, pages 189–204, 2015.

[14] Laird. S9028PCL. `https://www.lairdtech.com/products/s9028pcl`.

[15] Jia Liu, Xingyu Chen, Xiulong Liu, Xiaocong Zhang, Xia Wang, and Lijun Chen. On improving write throughput in commodity RFID systems. In *Proc. of IEEE INFOCOM*, pages 1522–1530, 2019.

[16] Xiulong Liu, Jiannong Cao, Yanni Yang, Wenyu Qu, Xibin Zhao, Keqiu Li, and Didi Yao. Fast RFID sensory data collection: Trade-off between computation and communication costs. *IEEE/ACM Transactions on Networking*, 27(3):1179–1191, 2019.

[17] Xuan Liu, Bin Xiao, Feng Zhu, and Shigeng Zhang. Let's work together: Fast tag identification by interference elimination for multiple RFID readers. In *Proc. of IEEE ICNP*, pages 1–10, 2016.

[18] Li Lu, Jinsong Han, Lei Hu, Yunhao Liu, and Lionel M Ni. Dynamic key-updating: Privacy-preserving authentication for RFID systems. In *Proc. of IEEE PerCom*, pages 13–22, 2007.

[19] Chen Min and Shigang Chen. ETAP: Enable lightweight anonymous RFID authentication with O(1) overhead. In *Proc. of IEEE ICNP*, pages 267–278, 2015.

[20] Senthilkumar Chinnappa Gounder Periaswamy, Dale R Thompson, and Jia Di. Fingerprinting RFID tags. *IEEE Transactions on Dependable and Secure Computing*, 8(6):938–943, 2010.

[21] Adam C Polak, Sepideh Dolatshahi, and Dennis L Goeckel. Identifying wireless users via transmitter imperfections. *IEEE Journal on Selected Areas in Communications*, 29(7):1469–1479, 2011.

[22] Saiyu Qi, Yuanqing Zheng, Mo Li, Yunhao Liu, and Jinli Qiu. Scalable data access control in RFID-enabled supply chain. In *Proc. of IEEE ICNP*, pages 71–82, 2014.

[23] Kristin Rasmussen. *Encyclopedia of measurement and statistics*, volume 1. Sage, 2007.

[24] Longfei Shangguan and Kyle Jamieson. The design and implementation of a mobile RFID tag sorting robot. In *Proc. of ACM MobiSys*, pages 31–42, 2016.

[25] Longfei Shangguan, Zheng Yang, Alex X Liu, Zimu Zhou, and Yunhao Liu. Relative localization of RFID tags using spatial-temporal phase profiling. In *Proc. of USENIX NSDI*, pages 251–263, 2015.

[26] Jitendra K Tugnait and Hyosung Kim. A channel-based hypothesis testing approach to enhance user authentication in wireless networks. In *Proc. of IEEE COMSNETS*, pages 1–9, 2010.

[27] Chuyu Wang, Lei Xie, Keyan Zhang, Wei Wang, Yanling Bu, and Sanglu Lu. Spin-Antenna: 3D motion tracking for tag array labeled objects via spinning antenna. In *Proc. of IEEE INFOCOM*, pages 1–9, 2019.

[28] Ju Wang, Liqiong Chang, Omid Abari, and Srinivasan Keshav. Are RFID sensing systems ready for the real world? In *Proc. of ACM MobiSys*, pages 366–377, 2019.

[29] Wenhao Wang, Zhi Sun, Sixu Piao, Bocheng Zhu, and Kui Ren. Wireless physical-layer identification: Modeling and validation. *IEEE Transactions on Information Forensics and Security*, 11(9):2091–2106, 2016.

[30] Teng Wei and Xinyu Zhang. Tracking orientation of batteryless internet-of-things using RFID tags. In *Proc. of ACM MobiCom*, pages 483–484, 2016.

[31] Liang Xiao, Larry Greenstein, Narayan Mandayam, and Wade Trappe. Fingerprints in the ether: Using the

physical layer for wireless authentication. In *Proc. of IEEE ICC*, pages 4646–4651, 2007.

[32] Lei Yang, Jinsong Han, Yong Qi, and Yunhao Liu. Identification-free batch authentication for RFID tags. In *Proc. of IEEE ICNP*, pages 154–163, 2010.

[33] Lei Yang, Pai Peng, Fan Dang, Cheng Wang, Xiang Yang Li, and Yunhao Liu. Anti-counterfeiting via federated RFID tags' fingerprints and geometric relationships. In *Proc. of IEEE INFOCOM*, pages 1–9, 2015.

[34] Jihong Yu, Wei Gong, Jiangchuan Liu, and Lin Chen. Fast and reliable tag search in large-scale RFID systems: A probabilistic tree-based approach. In *Proc. of IEEE INFOCOM*, pages 1133–1141, 2018.

[35] Davide Zanetti, Boris Danev, and Srdjan Capkun. Physical-layer identification of UHF RFID tags. In *Proc. of ACM MobiCom*, pages 353–364, 2010.

[36] Davide Zanetti, Pascal Sachs, and Srdjan Capkun. On the practicality of UHF RFID fingerprinting: How real is the RFID tracking problem? In *Proc. of Springer PETS*, pages 97–116, 2011.

[37] Wondimu K Zegeye, Seifemichael B Amsalu, Yacob Astatke, and Farzad Moazzami. WiFi RSS fingerprinting indoor localization for mobile devices. In *Proc. of IEEE UEMCON*, pages 1–6, 2016.

[38] Yan Zhang, Laurence T Yang, and Jiming Chen. *RFID and Sensor Networks: Architectures, Protocols, Security, and Integrations*. CRC Press, 2009.

[39] Tianhang Zheng, Zhi Sun, and Kui Ren. FID: Function modeling-based data-independent and channel-robust physical-layer identification. In *Proc. of IEEE INFOCOM*, pages 199–207, 2019.

[40] Anding Zhu and Thomas J Brazil. Behavioral modeling of RF power amplifiers based on pruned volterra series. *IEEE Microwave and Wireless components letters*, 14(12):563–565, 2004.