OpenTag: Privacy protection for RFID

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th>Citation</th>
<th>Holtzman, H., Sanghoon Lee, and D. Shen. “OpenTag: Privacy Protection for RFID.” Pervasive Computing, IEEE 8.2 (2009): 71-77. © 2009 Institute of Electrical and Electronics Engineers</th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1109/MPRV.2009.33">http://dx.doi.org/10.1109/MPRV.2009.33</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Tue Apr 02 05:19:17 EDT 2019</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/52328">http://hdl.handle.net/1721.1/52328</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>
Radio frequency identification technology\(^1\) is a powerful tool for pervasive computing, allowing people to easily tag objects so that a ubiquitous computing infrastructure can identify and track them in the physical environment. RFID tagging’s commercial utility has been recognized in supply chain, logistics, and retail settings, and the trend is to increase RFID tagging of goods at the time of manufacturing.

Ideas and prototypes for systems that use RFID-tagged items in a home setting abound. For example, the Magic Medicine Cabinet\(^2\) monitors pill bottles and guides the patient on dosage. A smart refrigerator can track its contents, warn the owner of expiration dates, and suggest menus. These are just two of the many examples where RFID can help effectively build a network of things.\(^3\) Within a decade, there will be a remarkable number of tags on everyday objects in the environment, rendering such object networks practical. Aid-  

Privacy advocates have raised many concerns about EPC adoption, not least because the tag’s size, chattiness, and automated functionality pose threats to consumer privacy. In response, lawmakers in several states have proposed legislation to limit RFID’s use or to at least require adherence to specific labeling and data-retention practices. The official trade-association solution is a kill command that disables the tag when the tagged item is purchased, but this renders the tag useless for all further functionality and usage.

To address privacy concerns and allow post-purchase tag use, we developed OpenTag, a semi-active, programmable RFID tag that uses the EPC protocol and passive communications. As we describe here, OpenTag uses hash-chains, a touch sensor, and a light sensor to ensure greater privacy for consumers and restore their control over how and when the tag transmits information.

RFID and Privacy

The Massachusetts Institute of Technology developed EPC as a comprehensive RFID tagging system for manufactured items. EPC has emerged as the industry standard for such tagging, having been adopted by Walmart, the US Department of Defense, the Federal Drug Administration, and many others to solve their RFID tagging needs.

Privacy advocates are intensely scrutinizing this EPC adoption, labeling both RFID and

---

Henry Holtzman, Sanghoon Lee,  
and Daniel Shen  
MIT Media Laboratory
EPC “spychip” technology and calling for a moratorium on their use in consumer products.4 There’s reason for concern. RFID is an inherently stealth technology: the tags operate automatically, without human intervention, and their operation is undetectable by human senses. EPC tags can be miniscule and completely hidden within products. Furthermore, EPC tags are chatty by nature, responding to any compatible reader that queries them. Similarly, RFID readers can work at long range and be invisibly integrated into the environment.

The EPC tag’s unique ID introduces a specific pair of privacy threats: it can be used both to track people and to inventory their possessions without permission. This electronic “frisking” is possible because EPC’s coding system not only uniquely identifies each tag but also encodes descriptions of the tagged items. When an EPC reader asks the tag for its data, the tag must respond to the request. So, any tagged item’s data can be exposed and the tag carrier has no control over how that data will be used. Readers can track a tag’s trail in time and space through the RFID network, as well as detect whether people are carrying particular—or particularly expensive—items.

EPCglobal, the EPC trade association, has attempted to address these privacy concerns at both technical and policy levels. Leaving aside the organization’s policy recommendations—which “bad actors” are unlikely to follow—its technical solution, a kill command, is problematic at several levels. When sent to an EPC RFID tag, the kill command will permanently disable it. Because a killed tag no longer responds to interrogation, it’s no longer a privacy threat. However, it’s also rendered useless as an RFID tag. According to current EPC standards, privacy and functionality are mutually exclusive.

Also, while killing tags eliminates the privacy risk, assuring that tags will be killed when items are sold to consumers is problematic. Killing a tag post-sale isn’t in a store’s best interest. Stores can use RFID tags to continue customer relationships, whether as part of a product-return process, to facilitate an in-home service call, or in conjunction with at-home RFID readers and applications. Thus, “kill” implies a set of business practices that, while privacy enhancing, reduce the RFID tag’s return on investment by eliminating further use. This inconveniences customers and burdens the store with additional check-out steps; it also requires that retailers have a way to replace tags if they’re returning products to inventory.

Finding a way to maintain RFID tag privacy and functionality outside of the supply chain could offer several key benefits. Such a privacy-enhanced tag could:

- enable consumers to inventory their own items for their own purposes,
- facilitate the continued relationship between a product’s manufacturer and consumer, and
- ensure proper disposal and recycling of goods at the end of their lives.

Active tags can also significantly increase the boundary of applications in a ubiquitous network. Motivated by a desire to build on and improve the EPC standard to this end, we developed the OpenTag RFID prototyping platform.

**Figure 1.** The OpenTag life cycle. The programmable RFID tag is compatible with EPC readers and tags, but includes enhanced privacy features.

**OpenTag**

OpenTag presumes that vendors will apply an RFID tag to a product at manufacturing and that it will remain affixed to that product throughout its life cycle (see Figure 1). OpenTag is compatible with EPC standards, which are the preeminent standards for source tagging goods. However, we’ve added enhanced privacy features and other functionalities to OpenTag, by creating a programmable RFID tag. Manufacturers have miniaturized, specialized, and reduced the cost of commercial RFID tags, making them unsuitable as a platform for developing new protocols, algorithms, and capabilities. By contrast, OpenTag is fully programmable.

We designed OpenTag to be compatible with EPC readers and tags, while incorporating privacy-enhancing features. In the supply chain, the OpenTag functions like any other first-generation EPC tag.5 However, once a consumer buys a product, rather than kill the tag, they use a private reader to set the tag in “private mode.” In this mode, the encrypted tag still responds to standard EPC readers using the EPC generation 1 protocol, but it gives different pseudo-IDs each time it’s read. A pseudo-ID is a nickname for the original, unencrypted response that the tag would have given in public mode. Readers that don’t share keys with the tag won’t be able to decode the pseudo-ID, making the tag unidentifiable and untraceable.

We’ve also developed a general framework for transferring between public and private modes and for transferring ownership. We demonstrate our system using OpenTagV2, an active, programmable tag that obeys the EPC protocol.
OpenTag V₂ comprises a 1-bit photosensor, a switch, and enough computation power to support light-weight cryptography.

**OpenTag Design and Implementation**

The features we implemented can't be tested on commercially available RFID tags as the chips available to create EPC-compliant tags aren't programmable. Also, they don't have sufficient computation power or expandable ports for such purposes. We’ve therefore built a series of custom RFID tags that combine an EPC RFID tag’s features with the features we require. We call these tags *OpenTag*.

**Design Evolution and Overview**

We built the first-generation OpenTag using an EPC RF modem board from TagSense⁶ and piggybacked it with a logic board to perform our special computations. Using this OpenTag, we demonstrated enhanced privacy using simple pseudonyms and a more secure, dynamically generated ID mode. This ID mode was based on a hash chain using a pseudo random number generator (PRNG) that revealed the algorithm while keeping the ID’s sequence predictable only to those who knew the secret key.⁷

Our current design—OpenTag V₂—eliminates the TagSense modem and has all of the necessary circuitry and computation on a single board. As Figure 2 shows, OpenTag V2 has an ATMega168 microcontroller with sufficient computation capability—20 million instructions per second (MIPS)—to perform hash algorithms and external sensors for advanced features.⁸

We programmed the EPC class 1 generation 1 protocol and security functions using assembly language for ATMEL. To communicate with our OpenTags, we used ThinMagic’s Mercury 4 RFID reader.

**Photosensor**

OpenTag V₂ uses Photonic Detector’s PDV-P5001 photoresistor to monitor the presence of light. We combine it with a voltage divider to detect whether the tag’s environment is brighter or darker than a threshold. The PDV-P5001 is sensitive to light from roughly 400 to 700 nanometers, which is the light range frequency visible to human eyes. For our application, the photosensor shouldn’t be infrared sensitive because infrared rays penetrate materials that humans can’t look through. The PDV-P5001 has a relatively slow response—a few tens of milliseconds—which gives a smooth response to jittery light sources.

**Reset Switch**

The tag’s switch (touch sensor) temporarily unlocks the tag from private mode. The switch is basically a two-terminal pad that activates when both pads are connected together with a conductive material. In most cases, connecting the two terminals with the pad of a finger will register, depending on the fingertip’s moisture. For increased reliability, we could use conductive foam.

**OpenTag’s ID Security**

We have three requirements for OpenTag’s ID security algorithm: It must be compatible with EPC (which transmits values “in the clear” from the tag to the reader), and it should protect the tag’s holder from both being tracked and being identified.

We initially implemented pseudonyms⁹ to prevent unwanted identification and location tracking. Our tag had a pseudonyms list and sent out different numbers each time the tag was read. We stored the pseudonyms in authorized readers so they could lookup and find the tag’s real identity. The advantage of using pseudonyms is that they’re easy to implement on a commercial tag.
The H parameter function uses the tag’s secret key to one-way transform its previous value \( V(n-1) \) into a new value \( V(n) \). The function \( G \) hashes \( V(n) \) to \( W(n) \) to keep \( V(n) \) secret, and generates \( W(n) \), which is emitted in place of the EPC value when the reader is in private mode. An authorized reader has both the electronic product code (EPC) and private key (the \( G \) and \( H \) parameters) for each tag it is tracking.

### Hash Duration

To defend its private key, the tag should limit how many pseudo-IDs it reveals to a reader. Although hash functions are forward secured, it’s still possible to narrow down the private key using brute force by calculating hash chains for all possible private keys and comparing these sequences to the tag’s replies. Revealing a smaller set of pseudo-IDs over a given time period helps secure the tag’s information. However, maintaining the same ID for a longer time also increases the tag’s trail. Although the tag shouldn’t give the same pseudo-ID to multiple readers adjacent to each other, the tag has no way to recognize the difference among readers.

We improve OpenTag’s security by using time and light to reduce the frequency of new pseudo-ID generation. Once a tag enters a reader’s range, it continuously receives energy for milliseconds (EPC gen1). By giving the tag time-keeping ability, it can recognize that it’s still in the same reader’s range if the interval between two received commands is less than a certain duration. If the tag powers up and sees that the last command it received was 1 second before, it has likely left the previous reader’s range and moved to a new reader, and it thus generates a new pseudo-ID. To handle cases in which readers are tightly coupled with overlapping fields, we added another timer: if a tag is in a field for a certain length of time such as 5 seconds without a break—the tag generates a new ID.

Our photosensor shuts down the tag’s activity in the dark. For example, if users keep their private tags in a bag, where it’s typically dark, the tags won’t reply to the reader. This lets the user
manually control the tag, preventing it from responding to readers and thus denying them the knowledge that the tag even exists. While it’s in the bag, the tag stops generating new pseudo-IDs; when it comes out it, it generates a pseudo-ID different from the one it showed before entering. This clearly secures the tag’s privacy and also slows down pseudo-ID generation.

**Evaluation**

Our goal is to show that the EPC system can be enhanced to provide privacy and functionality simultaneously. To function well, we evaluate how well the OpenTag prototype addresses security and if it is feasible to port our enhancements—from a cost and power perspective—into purely passive EPC tags.

**Security**

The hash scheme developed by Ohkubo and colleagues uses publicly known functions, H and G, but a different seed for each tag. Shingo Kinoshita and Ohkubo\(^\text{11}\) later demonstrated the hash chain scheme on an active RFID tag with an embedded 8-bit microprocessor. They implemented HA-1 and MD5 hash functions, which have acceptable security performances. Results show that one hash cycle took less than a second to calculate with their 20 MIPS 8-bit microprocessor.

Still, our goal with OpenTag is that our algorithms can ultimately be implemented in a purely passive tag, so we can’t adopt a scenario in which the tag is guaranteed to be consistently powered for a full second. A single hash process’s cycle should be short to avoid termination by abrupt power failures. If a hash function takes longer than a read cycle, the tag will be vulnerable to an attack using power scarcity; OpenTag can’t generate a new pseudo-ID if the reader tries to shut the power off right after a successful read. In the EPC class 1 generation 1 protocol, a read cycle is a few milliseconds, with less than 200 microseconds of idle time (not including the time a tag is communicating with a reader). Given its limited time and CPU speed, the system can’t fully support hash functions used in a common PC environment.

LCGs are extremely fast and have the least hardware requirements, but their security algorithm is weak compared to other common hash functions. We complemented the hash function’s weakness by setting an interval between hash cycles, randomly skipping past hash sequence values to complicate brute force pattern-matching attacks, and by providing physical (light and touch) control of tag activity. These steps enhanced OpenTag’s overall security. However, a better hash function is essential if OpenTag is to be acceptable to current standards; finding such functions are an outstanding challenge.

**Cost**

We use Verilog to simulate OpenTag’s additional functions; the system’s security functions require about 3.5K logic gates. The MIT Auto-ID Center proposed that a practical US$0.05 EPC tag design would be limited to between 7.5 and 15K gates. Given that an inexpensive 100-bit EPC tag requires approximately 5 to 10K gates,\(^\text{12}\) an additional 3.5K gates falls within the guidelines.

Our proposed sensors require more than just incremental logic; we also suggest additional components and external connections that will add to the total cost. We could embed a one-pixel complementary metal-oxide semiconductor sensor or a phototransistor into the chip as a photosensor, but it would affect the chip’s size and packaging.

**Power Consumption**

While passively coupled to the reader, the OpenTags we’ve built so far require battery assistance. Analyzing how much power the additional 3.5K gates and the sensors will consume isn’t straightforward. The two sensors require very little power because they can function with small bias currents. The reader’s signal strength, the tag’s distance from the reader, and the tag antenna’s dimensions determine how much power a tag can harvest. We could achieve the necessary power increase to support our functionality by trading off the working distance or the tag’s dimension. Also, using a smaller scale process for the chip would help decrease its overall power consumption. We expect that it’ll soon be possible to create a passively powered chip with our tag’s increased functionality without negatively effecting distance or size as compared to today’s tags.

As further evidence that a passive tag with additional functionality is possible, researchers have demonstrated a wirelessly powered platform\(^\text{13}\) and a
passive sensor-integrated RFID tag that can use energy harvested from the RFID channel to enable additional sensing and computations.

**User Scenario**
As Figure 4 shows, OpenTag currently has three operational modes: public, private, and invisible.

**EPC Mode**
In public, or EPC mode, the tag acts like a normal EPC tag, sending its code without privacy enhancement. EPC mode has two states: locked and unlocked. Locked EPC mode is for OpenTags in the supply chain; in this mode, privacy features can’t be activated. When a consumer buys the product from a store, the OpenTag is unlocked. Only an authorized RFID reader can unlock the OpenTag using a pre-programmed password. When users bring home an OpenTag, their household readers would automatically load it with the household’s private key. It’s possible that an OpenTag could get associated with a nearby unknown reader en route to the target household. However, it’s easy to restore such a tag to unlocked EPC mode by touching the OpenTag’s switch. If this happens to numerous tags, it could be tedious work. To address this, a store could load the customer’s private key on the tags at checkout, deriving the key, for example, from the customer’s credit card number in a standard fashion. Each user’s readers could be programmed to know the private keys corresponding to each of the user’s credit cards. If customers are purchasing gifts or shopping for others, the store could program the tags with a random private key and print it on a gift receipt. We could even implement OpenTag to privatize key and print it on a gift receipt. Disposal and Changing Ownership
A tag’s information is valuable in classifying recyclable materials at disposal. To be properly sorted at a disposal facility, however, the tag must be switched back to EPC mode. Users can do this by simply disassociating the tag from the reader. However, some users might forget to do this. Also, there are many other situations when it might be desirable to exit OpenTag’s private mode, such as when you’re returning a product to a shop, identifying items at a lost and found, or receiving a second-hand item from eBay.

Therefore, we must include a way to manually break the private mode and restore the tag to EPC mode. Because offering a master key to break the private mode could threaten the security of OpenTag’s private mode, we introduced a physical reset method. A public reader can set the tag to EPC mode when a switch on the tag is touched. The switch consists of two small metal pads that are a few millimeters apart. The switch is triggered when a finger or conductive material connects the two terminals. Because users will rarely activate the switch, we could initially cover the terminals with a thin film mask.

**Private Mode**
Once a user buys the item, the tag is ready to be associated with a private key so that only authorized readers can track it. The tag is open to any request—as with a new laptop computer, the first person to set the password locks it for their private use. Likewise, the gift’s recipient could then enter the private key into their own reader, which would automatically reprogram all of the items received.

**Invisible Mode**
Although we secure the tag’s identity with pseudo-IDs, receiving a signal—even if it’s disguised—still tells nearby readers that an item is present. We therefore added an invisible mode that stops the tag from replying to all requests.

When people want to secure things, they typically hide them out of sight, often in dark places such as pockets, backpacks, and boxes. We used this same notion for OpenTag: it goes into invisible mode when its photosensor detects light that’s lower than a certain threshold. The mode electronically blocks communication between the tag and readers, ensuring high security.

Also, in reader-dense environments, keeping tags in the invisible mode prevents them from exposing their pseudo-IDs. This mode can be enabled or disabled when a private reader programs the tags. Also, users can temporarily disable an OpenTag by simply placing a piece of opaque tape over the photosensor.

**Disposal and Changing Ownership**
A tag’s information is valuable in classifying recyclable materials at disposal. To be properly sorted at a disposal facility, however, the tag must be switched back to EPC mode. Users can do this by simply disassociating the tag from the reader. However, some users might forget to do this. Also, there are many other situations when it might be desirable to exit OpenTag’s private mode, such as when you’re returning a product to a shop, identifying items at a lost and found, or receiving a second-hand item from eBay.

Therefore, we must include a way to manually break the private mode and restore the tag to EPC mode. Because offering a master key to break the private mode could threaten the security of OpenTag’s private mode, we introduced a physical reset method. A public reader can set the tag to EPC mode when a switch on the tag is touched. The switch consists of two small metal pads that are a few millimeters apart. The switch is triggered when a finger or conductive material connects the two terminals. Because users will rarely activate the switch, we could initially cover the terminals with a thin film mask.

**The invisibility mode electronically blocks communication between the tag and readers, ensuring high security.**

We’ve used batteries to power the two OpenTag versions we’ve built thus far and coupled them with readers. A natural question is whether our proposed privacy methods are applicable to purely passive tags. After examining the possible increase in cost and power consumption for the...
security features, we've concluded that EPC passive tags could reasonably support them. We therefore suggest that EPC consider replacing the kill function with a privacy enhancement mode such as ours to increase the EPC system's security and usefulness.

Our future work with OpenTag will embrace its programmability and semi-active nature to explore how we can further use tags that include sensors. Typically, RFID readers can determine if a tag is within their field, but know little about the tag's state. Using our light and touch sensors—as well as tilt and possibly other sensors—our next version will sense aspects of its environment and return these results to the RFID reader.

REFERENCES


8. 8-bit AVR Microcontroller with 8K Bytes In-System Programmable Flash ATmega48/V, ATmega88/V and ATmega168/V, Revision 2545F- AVR-06/05, Atmel, 2005; www.atmel.com/dyn/resources/prod_documents/doc2545.pdf.


For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.