The Design and Assessment of a Secure Passive RFID Sensor System

Michael Todd, Wayne Burleson, and Russell Tessier
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA, USA

Abstract—**This paper presents a low-overhead security enhancement for EPC Class-1 Generation-2 (Gen2) compatible RFID tags that provides data confidentiality for a series of common threats. The new security circuit, based on the PRESENT block cipher, is fully integrated into a passive RFID tag architecture. The circuit is evaluated in an FPGA-based emulation platform which allows for the validation of the circuit and the use of a protocol which seamlessly interacts with a standard off-the-shelf RFID reader. A complete system, including a temperature sensor attached to the emulated tag, was successfully tested in the lab using an existing Gen2 RFID reader. The hardware overhead of the security enhancement is roughly 1,900 logic gates.**

I. Introduction

Passive Radio Frequency Identification (RFID) transponders (tags) are low-cost, low-power integrated devices which harvest RF energy to power a small amount of analog circuitry, digital logic, non-volatile memory, and an antenna. A typical RFID system consists of a host computer, RFID interrogator (reader), and tag (Figure 1). Passive tags do not have a battery or other external energy source, necessitating a low-power and low-complexity design. A backscatter process is used to reflect RF energy transmitted to the tag back to the reader. RFID devices have been deployed in several industries including military, pharmaceutical, and retail sales [1].

The EPCglobal Class-1 Generation-2 Radio-Frequency Identity Protocol for Communications at 860 MHz – 960MHz, or Gen2, has been widely accepted as the de-facto standard for passively-powered RFID tags since its inception by EPCglobal in 2004 [2] and the International Organization for Standardization (ISO) 18000-6C in 2006. Recently, the application space of passive Gen2 tags has expanded to include serving as a front end for simple environmental sensors, such as thermal sensors and heart monitors. For many applications, transferred data is sensitive, necessitating data security within the tight power and area constraints of standard passive tags. To maintain continuity, any new data transfer protocol must maintain compatibility with the sizable base of existing Gen2-installed readers that are deployed around the world.

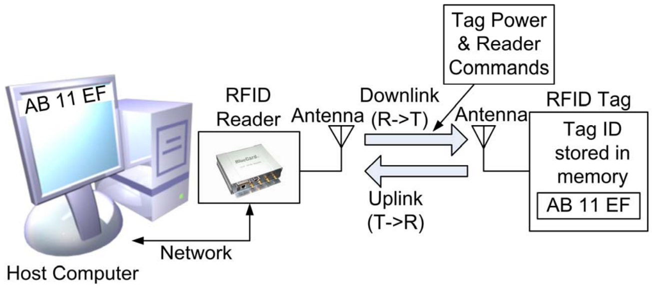

Figure 1. RFID infrastructure including host computer, RFID reader, and RFID tag

This work explores the design, validation, and emulation of a Gen2 compliant secure environmental sensing RFID platform. A Gen2 tag architecture has been augmented with a recently-developed block cipher, allowing for tag data transmission security. A secure reader-tag communication protocol is introduced which is fully compatible with existing Gen2 readers and can be implemented as part of standard Gen2 communication transactions. Our new security architecture and protocol have been validated in hardware with a WISP [3] RFID front end, FPGA emulation of tag hardware, and a Gen2 compliant tag reader. A downlink data rate of up to 215 Kbps and an upload rate of up to 640 Kbps are supported.

II. Background

A standard Gen2 reader controls several aspects of each reader-tag communication session such as downlink (R→T) data rate (between 50-215 kbps), and uplink (T→R) data rate (between 5-640 kbps). Tags must support a basic set of commands, although space is reserved for custom commands. Tags are also assumed to have some form of non-volatile memory such as flash, to store the Electronic Product Code (EPC_{ID}), which can be thought of as the ID of the tag, as well as other information.

The basic building blocks of a standard Gen2 RFID tag are depicted in Figure 2. The two primary modules of the tag are the analog frontend and digital backend. The frontend is responsible for regulating the incoming RF signal to generate the supply voltage for the digital logic, generating a clock signal, and demodulating the incoming data. The frontend also contains the backscatter transistor used to alter the reflection coefficient of the antenna during T→R communication. This paper concentrates on the Digital

This work has been partially sponsored by the National Science Foundation grant numbers CNS-0627529, CNS-0923313, CNS-0964641

Components & Control Logic portion. An expanded view of the digital backend required in a standard Gen2 compliant tag includes a number of blocks. The basic blocks consist of a downlink pulse interval encoding (PIE) data decoder, a backscatter clock divider which generates the uplink clock between 40-640 kHz, an uplink data encoder that must support both FM0 and Miller encoding schemes, 5-bit and 16-bit cyclic redundancy checks, a 16-bit pseudo random number generator (PRNG), memory controller, and a command handler finite state machine.

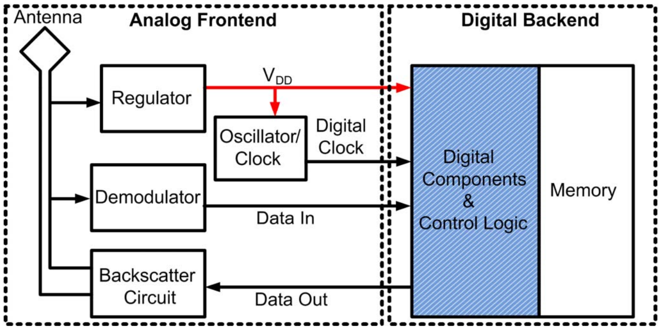

Figure 2. The basic components of an RFID tag.

III. Secure Sensing with A Gen2 Tag

The expanded use of RFID sensors in medical, military and infrastructure maintenance applications motivates the development of a low-overhead method to securely transmit RFID sensor data. To illustrate our approach, a detailed scenario is presented. As an example, several sensors with passive RFID frontends could be embedded in a bridge. To minimize complexity, a vehicle equipped with an RFID reader periodically passes over the bridge and collects sensor data. The vehicle then returns to a location with secure access to a database and downloads the information for secure processing. With this example of separated data collection and later processing in mind, a threat model and security protocol is developed.

This scenario raises several security concerns. It is undesirable to directly reveal the sensor data for fear of an adversary using this information to target weak points on the bridge. One could also envision an attack in which previous sensor data is collected by the adversary and replayed to the trusted reader to hinder maintenance. Finally, it is desirable to avoid *indirectly* revealing information about the sensor by ensuring that given the same challenge, the response from the tag appears to change thus blocking the adversary from inferring the state of the sensor. Our threat model makes the following assumptions about an adversary: (1) The adversary can change any information traveling between the reader and the tag (untrusted reader or tag), (2) The adversary knows everything about the protocol we intend to use, (3) Each tag contains an 80-bit secret key that is unknown to the adversary, and (4) The adversary can request the data from the tag an unlimited number of times.

Since, traditionally, the primary intended function of Gen2 tags has been low-cost and high-volume inventory tracking, tag security is virtually nonexistent. While the standard Gen2 protocol does contain the transfer of a 16-bit value generated on the tag by a PRNG, the value mainly serves in an access control role, allowing tags to verify the intended recipient of reader commands, as well as slotting tags in a response queue. Increased Gen2 security has been the focus of several research papers. In Man et al. [4], an Advanced Encryption Standard (AES) block cipher was added to a Gen2 tag at the cost of a large increase in tag area overhead. This approach also has a significant limitation regarding tag identification. Since all information from the tag is encrypted, including the EPC_{ID}, it can be difficult for a reader to identify the target tag if multiple tags respond to a query. The second drawback is the vulnerability to replay attacks. Bailey and Juels [5] proposed an alternate security scheme which addresses the replay issue for specific sensor values but not the tag's ability to mask changes in sensor data. Before responding to a query, a tag computes value $R_T = H(K_{TS}, C_R)$ where H() is a block cipher, K_{TS} is a shared secret key stored on the tag and reader (or backend server), and C_R is a unique challenge sent from the reader to the tag. The tag then replies with its EPC_{ID} and R_T. The major drawback to this scheme is that if C_R is static, R_T is also static.

IV. Enhanced Digital Backend

Before presenting the enhanced tag architecture, we describe the enhanced protocol which is used for secure reader-tag communication. The protocol is based on the recently-developed PRESENT block cipher [6] which includes an 80-bit key and a 64-bit data size. It satisfies the following constraints: (1) Full backwards compatibility with the Gen2 standard, (2) No uplink between reader and backend key database required during data collection, (3) Defended against replay attacks, (4) One way authentication of the data origin, and (5) Tag data integrity check. To keep system costs down, it is desirable to use a standard off-the-shelf reader that does not support encryption or decryption within the unit itself. Rather, encrypted data is collected locally (online) and then decrypted *remotely* (offline) at a location with a secure database, for example, after the truck in the bridge example has returned to its base.

Figure 3 represents the Reader-Tag online portion. RN16 is a 16-bit pseudo random number generated by the tag at power up. Upon receiving a QUERY command, the tag backscatters RN16. The reader then responds with an acknowledgement ACK command which contains RN16. After the tag confirms the validity of RN16, EPC_{ID} is backscattered. In our extension, once the EPC_{ID} has been received by the reader, the reader emits a “Data Request” command (derived from the Gen2 custom command opcode space [2]) and an 80-bit true random number (TRN_{80}). This number must be 80 bits to match the key size of the PRESENT block cipher. The tag then XORs TRN_{80} with the stored symmetric key (k_i), concatenates a 40-bit binary counter value (CV_{40}) with the 24 bits of sensor data (D_{24}), and encrypts it (e_{kj}) using the result of the 80-bit XOR operation to form the 64-bit encrypted result (C). This result and unencrypted version of the counter value (CV_{40}) is sent to the reader.

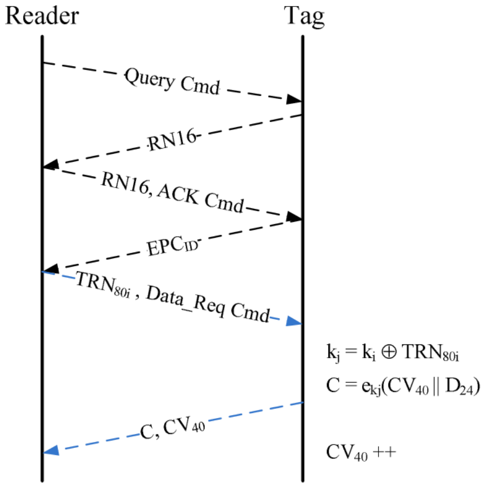
Reader
Tag
Query Cmd
RN16
RN16, ACK Cmd
EPC_{ID}
TRN_{80i} , Data_Req Cmd
$k_j = k_i \oplus TRN_{80i}$
$C = e_{kj}(CV_{40} \| D_{24})$
C, CV_{40}
CV_{40} ++

Figure 3. R-->T online portion of sensor data collection protocol.

Figure 4 depicts the second portion of our scheme, the offline data download. After all tags have been queried, the reader returns back to a secure location and uploads the tag information to the symmetric key database. Using the unencrypted $EPC_{ID,}$ the server is able to securely retrieve the appropriate shared symmetric key (k_i). Then, the server can perform an XOR between the key and TRN_{80} and decrypt (d_{kj}) the ciphertext C. The server can then determine the validity of C by checking if the decrypted version of CV_{40} matches the received version. The reader then receives the next TRN_{80} from the server. This removes the need for the reader to carry a true random number generator (TRNG). The reader will need a separate, unique TRN_{80} for each tag it plans to query.

The security protocol hinges upon the availability of a low overhead symmetric block cipher. PRESENT is a 64-bit block cipher with an 80-bit key, designed for low-power, low-gate count applications [6]. PRESENT is currently under review for ISO standardization as a low-power block cipher. The authors claim an approximate gate equivalent count of 1.5k in their implementation of PRESENT. The security of the protocol described above is derived from unilateral authentication using true random numbers. Further information regarding the security of the protocol is described elsewhere [7].

V. Architecture Implementation

To validate our new protocol, the architecture described in the previous section has been prototyped and tested with a standard Gen2 reader. Except for the PRESENT security core and temperature sensor interface, our tag implementation uses accepted techniques for Gen2 tag clocking and pseudo random number generation. A more detailed view of digital circuitry and temperature sensor appears in Figure 5.

Although a temperature sensor is used for validation in our prototype implementation, any low bandwidth sensor could be used in practice. For the bridge sensor example, a stress, strain, or humidity gauge could be used in addition to or in place of the temperature sensor.

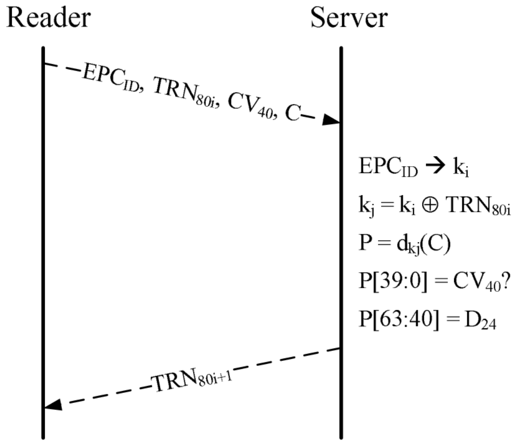
Reader
Server
EPC_{ID}, TRN_{80i}, CV_{40}, C
$EPC_{ID} \rightarrow k_i$
$k_j = k_i \oplus TRN_{80i}$
$P = d_{kj}(C)$
$P[39:0] = CV_{40}$?
$P[63:40] = D_{24}$
TRN_{80i+1}

Figure 4. Offline portion of sensor data collection protocol.

The Wireless Sensing Platform (WISP), designed by Intel [3], serves as the analog interface for our tag emulation platform. WISP is constructed from analog frontend circuitry, a low power TI MSP430 microcontroller, an accelerometer, and a temperature sensor. The *data in* and *data out* signals on the left side of the block diagram in Figure 10 directly interface to the WISP. For this system, the WISP is only used for analog interfacing, even though it contains a microcontroller. The operations required by the PRESENT algorithm were developed with hardware implementation in mind. It has previously been shown [6] that the algorithm is more efficiently implemented in terms of size and performance as a custom hardware design versus a software algorithm. Thus, the PRESENT algorithm is prototyped in hardware rather than using the WISP microcontroller.

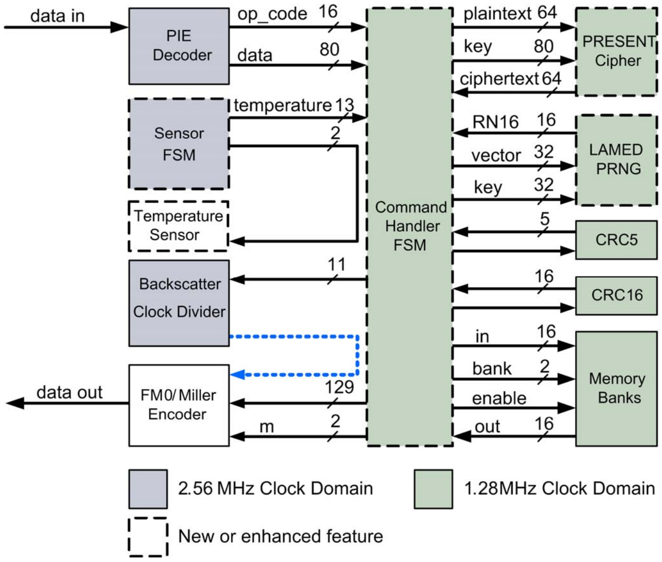

Figure 5. Block diagram of the enhanced Gen2 digital backend

LAMED is a pseudo random number generator first described in [8] which is used to conform to the RN16 generation requirements outlined in [2]. The LAMED algorithm is constructed from 32-bit XORs, 32-bit ADDs and variable length barrel shift operations. The simplicity of these operations keeps the design power requirements low. A separate clock is used to drive the PIE decoder and backscatter link frequency divider due to both the lower bit error rate and power consumption of the tag. The standard clock frequency quoted for driving the digital logic in a Gen2 tag is 1.28 MHz. Our design uses 2.56 MHz for the PIE decoder, backscatter clock divider, and temperature sensor FSM. The PRESENT cipher block was specified in RTL code and synthesized to hardware. The block was synthesized separately from the other cores in the design.

VI. Design Validation

A. Experimental Setup

Both the design and validation portions of this work were evaluated using circuits and testbenches written in the Verilog Hardware Definition Language (HDL) with compilation and simulation done in the Xilinx ISE 9.2.04i environment. The goal of the validation was to assess the new protocol and the practical implementation of logic components of the protocol in an emulated RFID system. One significant challenge was verifying that our system works to specification. Verilog simulations done in the Xilinx ISE environment provided initial verification. As the sub blocks were integrated into larger and more complex modules, constructing meaningful testbenches became more challenging necessitating in-circuit testing with the RFID reader. To solve this challenge, an Impinj Speedway RFID reader was used to test several aspects of our design including command decoding, backscatter encoding, backscatter frequency, and R→T symbol length (Figure 6).

The use of a 3rd party RFID reader allowed a qualitative and quantitative assessment of the performance of the design. The reader makes use of the opcode space reserved for custom commands to transmit our Data_Req command and random numbers. We emulated our protocol using only basic Gen2 commands. A Tektronix DPO7104 digital phosphor oscilloscope was used to probe our design and capture I/O traces. In addition to successfully sending and receiving ciphered data at Gen2 speeds using the reader and experimental setup, the behavior of critical signals was observed to verify correctness. During debug we were able to probe four signals with the oscilloscope. These signals were subsequently converted to a Verilog testbench. The testbench was then simulated using ISE tools. This process allowed us to conduct functional simulations with stimuli provided by the RFID reader and compare expected against actual results, greatly enhancing our understanding of the protocol as well as reducing debug time.

In addition to synthesizing the design to FPGA logic for implementation on the FPGA-based board, the design was synthesized with the Synopsys Design Compiler using the UMC 65nm standard cell library. Synthesis gate counts were then normalized to NAND2 gate equivalents (GE) to allow for comparison to previous implementations. The design contains approximately 16.7k GE. The PRESENT block size of 1,900 gates is similar to the value which has been reported in previous work [6]. The current prototype implementation of the design, which is highly modularized, could be further optimized to reduce its overall GE count. In particular, the Gen2 protocol control logic, which currently measures 9,700 GE, could be reduced with additional state machine optimization. The 16.7k GE figure is roughly similar to the 20.0k GE result [4] determined for an AES-based passive tag implemented 0.18 um technology.

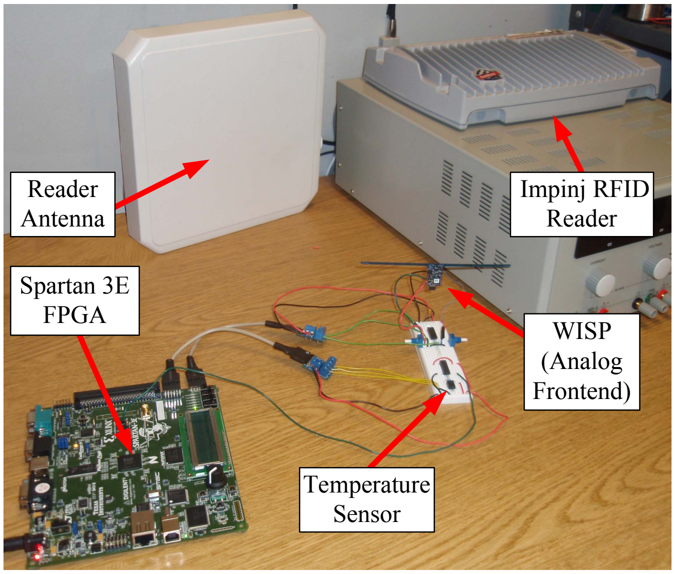

Figure 6. Photograph of the RFID emulation platform

VII. Conclusions

This paper presents a low-overhead architecture enhancement for Gen2 compatible RFID tags that provides data confidentiality for a series of common threats. The new security circuit is based on the PRESENT block cipher, a recently-developed, low-overhead block cipher. We have successfully evaluated an implementation of our security protocol and architecture using an FPGA-based emulation platform which seamlessly interacts with a standard off-the-shelf RFID reader.

References

[1] Mark D. Dobkin, The RF in RFID: Passive UHF RFID in Practice. Elsevier / Newnes, Amsterdam, 2008

[2] EPCglobal IncTM, EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz -960MHz Version 1.2.0 , October 2008.

[3] S. Ahson, and M. Ilyas. "Wisp: A Passively Powered UHF RFID Tag with Sensing and Computation." RFID Handbook: Applications, Technology, Security, and Privacy. CRC Publishers, Boca Raton, Fl, 2008.

[4] A. Man, et al., Low Power VLSI Design for a RFID Passive Tag Baseband System Enhanced with an AES Cryptography Proceedings of the RFID Eurasia Conference, 2007

[5] D. Bailey and A. Juels. "Shoehorning Security into the EPC Standard." *Security and Cryptography for Networks*. Vol. 4116. Lecture Notes in Computer Science, Vol. 4116, 303-320, 2006

[6] A. Bogdanov, et al. "PRESENT: An Ultra-Lightweight Block Cipher." Lecture Notes in Computer Science, 2007.

[7] M. Todd, Hardware Emulation of a Secure Passive RFID Sensor System, MS thesis, Department of ECE, UMass, Amherst, 2010.

[8] P. Peris-Lopez, et al. LAMED-A PRNG for EPC Class-1 Generation-2 RFID specification, Computer Standards and Interfaces, 2007.