

Maximalist Cryptography and Computation on the WISP UHF RFID Tag

Hee-Jin Chae¹, Daniel J. Yeager², Joshua R. Smith², and Kevin Fu¹

¹ University of Massachusetts, Amherst, MA, USA, {chae, kevinfu}@cs.umass.edu

² Intel Research Seattle, Seattle, WA, USA, yeagerd@ee.washington.edu,
Joshua.r.smith@intel.com
<http://www.rfid-cusp.org/>

Abstract. With continuous improvements in the efficiency of microelectronics, it is now possible to power a general-purpose microcontroller wirelessly at a reasonable range. Our implementation of RC5-32/12/16 on the WISP UHF RFID tag shows that conventional cryptography is no longer beyond the reach of a general-purpose UHF tag. In this paper, (1) we provide preliminary experimental data on how much computation is available on a TI MSP430F1232 microcontroller-based RFID tag containing approximately 8 KBytes of flash and 256 bytes of RAM, and (2) we show that symmetric cryptography is feasible on an RF-powered, general-purpose RFID tag — providing the first implementation of conventional cryptography on an RF-powered UHF RFID tag as far as we are aware.

1 Introduction

Because of computational constraints on many RFID tags, classical cryptographic primitives such as block ciphers and asymmetric cryptography were thought to be unrealistic on a low-resource tag [9]. To this end, many lightweight cryptographic protocols have been proposed [5, 6, 14–16, 18, 19]. However, many such protocols have serious vulnerabilities [7, 11, 12]. Moreover, the lack of a development platform makes it difficult to determine the feasibility of proposed cryptographic schemes. Thus, a popular approach is to minimize cryptographic operations to ensure feasibility on an RFID tag. This minimalist approach [9] can leave spare computational resources unused. An open question is then how to best maximize the security on an RFID tag to fully utilize available computational resources.

Our approach to maximizing security relies on low-power microcontrollers. Continuous improvements in efficiency of microelectronics (i.e., required energy per instruction) now enables wirelessly-powered, general-purpose microcontrollers — infeasible at any reasonable range a few years ago. Such microcontrollers make traditional cryptographic methods more feasible on RFID tags.

Using a maximalist approach to cryptography, our results show that an RF-powered UHF tag can perform strong encryption. We provide preliminary experimental results of implementing RC5-32/12/16 [22] on WISP (Wireless Identification and Sensing Platform) [24] — a battery-free platform powered and read

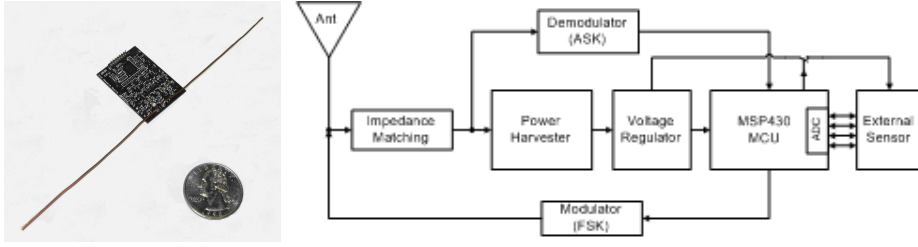


Fig. 1. WISP [24, 23] is a batteryless, microcontroller-based UHF tag that implements RFID protocols in software. WISP uses a TI MSP430 microcontroller and minimal analog circuitry to function as a UHF RFID tag.

by a standards compliant UHF RFID reader running the EPC Class 1 Gen 1 protocol. Our contributions include:

1. We provide preliminary experimental data on how much computation is available on a microcontroller-based RFID tag.
2. We show that symmetric cryptography is feasible on an RF-powered, general-purpose RFID tag. To the best of our knowledge, this is the first implementation of conventional cryptography on an RF-powered UHF RFID tag.

2 Background on WISP Architecture

This section provides a condensed background on the WISP, more fully described in [24, 23]. The WISP and its block diagram appear in Fig. 1. An antenna and impedance matching circuit precede the analog front end. The power harvester block rectifies incoming RF energy into DC voltage to power the system. The demodulator follows the envelope of the RF carrier wave to extract the Amplitude Shift Keyed (ASK) data. This extracted baseband waveform is read by the MSP430 microcontroller to receive downlink data from the reader. Uplink data is sent via the modulator circuit, which functions by changing the antenna impedance. The WISP is made of a two layer FR4 PCB with components limited to the top side. A dipole antenna made of 22 gauge (0.6mm diameter) copper magnet wire is visible. Small header pins expose all ports of the microcontroller for expansion to daughter boards, external sensors and peripherals.

Analog Front End and Tuning. Due to the relatively high power consumption of WISP, its rectifier is designed to supply more current than ordinary tags. This circuit is excited by commercial, EPC Class 1 Generation 1 compliant readers operating at 902-928 MHz with an allowable transmission power of $4W_{\text{EIRP}}$ (Effective Isotropic Radiated Power).

Efficient conversion of the incoming RF energy to DC power for the tag maximizes range. A matching network provides maximum power transfer from the antenna to the rectifier, and a 5 stage voltage doubling circuit converts the

incoming power to voltage. Low threshold RF Schottky diodes maximize the voltage output of the rectifier. Finally, this rectified DC voltage is stored in a large capacitor and supplied to a 1.8 V regulator to power the WISP.

Demodulation and Modulation. To encode reader-to-tag data, the reader amplitude-modulates the 915MHz RF carrier wave it emits. Normally the carrier waveform remains at a constant amplitude; when bits are transmitted, the amplitude of the carrier drops to approximately ten percent of its normal value. The duration of the low “break” indicates a logical “one” or a “zero.” A short break (1.78 μ S) indicates a “zero,” and a long break (5.34 μ S) indicates a “one.” To decode this data, the RF signal is fed through a small (two stage) voltage doubling rectifier in parallel with the main (five stage) harvester. We call the two stage harvester the “mini-harvester.” The time constant of the mini-harvester is much smaller than that of the main harvester, allowing it to track the dynamic range of the incoming bits. The first two voltage doubling stages of the mini-harvester, in conjunction with a lowpass filter, effectively demodulate the 915MHz carrier, and leave a baseband data signal on the order of 70kHz. A final “extra” diode performs an additional rectification step, removing the 70kHz data signal and leaving a slowly varying average power level (i.e., just fast enough so that it can change on the timescale that the tag moves in space, say 10Hz) that provides a dynamic reference for bit detection.

The 70kHz data signal is fed through a Schmitt trigger inverter that thresholds this waveform to remove noise and glitches. Finally, a level shifter converts the relative magnitude of the incoming data waveform into a 1.8 V logic level for the MSP430. The slowly varying average power signal serves as the power supply for the Schmitt trigger and level shifter.

RFID tags do not actively transmit radio signals. Instead they modulate the impedance of their antenna which causes a change in the amount of energy reflected back to the reader. This modulated reflection is typically called backscatter radiation. In order to change the impedance of the antenna, a transistor is placed between the two branches of the dipole antenna. When the transistor conducts current, it short circuits the two branches of the antenna together, changing the antenna impedance; in the non-conducting state, the transistor has no effect on the antenna, and thus the power harvesting and data downlink functions occur as if it were not present. This impedance modulation is currently implemented with a 5GHz RF bipolar junction transistor which allows for effective shunting of the 915MHz carrier wave.

Digital section and power conditioning. The WISP’s general-purpose computation capabilities are provided by an ultra low power microcontroller, the TI MSP430F1232. This 16-bit device has just over 8 KBytes of flash memory, 256 bytes of RAM and a 10-bit, 200kilo-samples-per-second Analog to Digital Converter (ADC). It can run at 8MHz with a 3.3 V supply voltage, 6MHz with a 3.0 V supply, and 4Mhz with a 1.8 V supply. To erase and write to flash memory, a minimum 2.7 V supply is required. In this application, we used a 3.0 V supply because a suitable ultra-low quiescent current 2.7 V regulator was not available.

In active operation at 3.0 V, the microcontroller consumes approximately 690 μ A to 1.4 mA. Erasing and writing flash both require approximately 3mA. Reading from flash requires no additional power (above ordinary active-mode operation). The flash in the MSP430's information memory must be erased in 128 byte blocks; its main memory is erased in 512 byte blocks. Bytes are individually writable. Erasing the flash memory takes more time and therefore more energy than writing or reading flash. Erasing a block requires 1.51 ms; writing a byte takes 32.7 μ s; reading a byte from flash requires no more time than accessing RAM.

The MSP430 has various low power modes that are very useful for wirelessly powered operation. Its lowest-power mode, "RAM-retention mode," which stops all computation but maintains state, requires only 0.5 μ A at 1.5 V. The low power consumption of this relatively new device is a critical factor in enabling use of a general-purpose microcontroller in RF-powered RFID systems.

WISP and EPC Gen 2. The WISP team is planning to add support for the EPC Class 1 Gen 2 standard. The Gen 2 physical layer and state-based protocol is more complex than Gen 1. This will require an increase in processing and memory. However, we believe that the WISP approach can be extended to implement this more complex protocol.

3 Related Work

Because of the resource-constrained nature of nodes in sensor networks, many of the design criteria for security coincide with that for security of RFID tags. SPINS [17] and TinySec [10] both present experimental data on implementation of security protocols in sensor networks. Both works implemented RC5 for their block ciphers because of its small code size and high efficiency, and they show that RC5 provides a balance between security and performance for sensor networks. Although sensor nodes have limited resources, general-purpose RFID tags are even more limited in memory, power, and computing capabilities. Sensor nodes like the Mica2 platform have their own power source onboard, and they have orders of magnitude more storage. Table 3 provides side-by-side comparison of different devices with RF interfaces. WISP is most comparable to general-purpose RFID tags such as EPC Gen1 and Gen2.

There are many HF tags capable of cryptography because of their common usage in building access control and authentication applications [13]. HF tags can afford to be more expensive in terms of manufacture cost and therefore provide stronger cryptography because the demand for tags with built-in security and privacy mechanisms, which is usually achieved with a separate cryptographic engine. Israsena proposes the Tiny Encryption Algorithm (TEA) as a suitable encryption engine for low-cost RFID applications [8]. Three different architectures for the TEA encryption algorithm are presented that fit within the budget of a 5 cent per tag cost. Aigner and Feldhofer discuss their results on an implementation of Tiny AES (TINA) as an ASIC [2]. Although both systems comply with

| Platform | Power | Computing | Storage | Communication | Distance |
|-------------------|-----------|---------------|------------------------------------|-----------------------|------------------|
| EPC Gen1 [20] | UHF RF | State machine | 64 bits | UHF backscatter | 3-7.5 m [26] |
| EPC Gen2 [20] | UHF RF | State machine | 96/128 bits | UHF backscatter | 3-7.5 m [26] |
| WISP [23] | UHF RF | 16-bit 8MHz | 8KB+ 256 bytes flash 256 bytes RAM | UHF backscatter | < 4.5 m |
| DemoTag [1] | Battery | 8-bit 16MHz | 4KB EEPROM 4KB SRAM 128KB flash | HF backscatter | N/A |
| Mica2 | Battery | 8-bit 8MHz | 128KB ROM 4KB RAM 512KB flash | UHF/ 433MHz | 40-50 m [3] |
| Microchip MCRF202 | Inductive | 12-bit 400kHz | 96/128 bits | LF backscatter | 1.3-10.2 cm [25] |
| Proxmark3 [27] | USB | 32-bit 16MHz | 256KB flash 64KB SRAM | LF and HF backscatter | N/A |
| RFIDGuardian [21] | Battery | 32-bit 520MHz | 16MB flash 64MB SRAM | HF backscatter | .5 m |

Table 1. Comparison of WISP with other RFID devices.

stringent requirements for low-cost RFID systems (average current consumption below 10 μ A), they depend on separate special-purpose circuitry. While ASICs are efficient in terms of power consumption and cost, they are inflexible and limited to a narrow set of applications. Because of this inflexibility, the cost for prototyping and development is relatively high — making it difficult and time consuming to realize in actual applications. To the best of our knowledge, our system provides the first UHF RFID tag using general-purpose microcontroller to provide cryptographic capabilities under non-trivial computational constraints.

4 Measurements and Estimates of Computation and Power Consumption

In this section, we demonstrate that symmetric cryptography is feasible on microcontroller-based RFID tags through an empirical study of RC5-32/12/16 on the WISP. Experimental data shows how much computation is available on WISP with varying workloads. Although WISP is relatively power-inefficient

compared to ASIC designs, the logic gate count of WISP is comparable to an EPC UHF tag — enabling a reasonable measurement of computational capabilities and limitations of an RFID tag.

4.1 Measurements of Computation and Symmetric Cryptography

To show the feasibility of classical cryptography on a general-purpose RFID tag, we present experimental results of implementing RC5 on WISP. We have chosen RC5 because of its simplicity and relatively small memory requirements. RC5 is implemented with 32-bit word, 12 rounds, and 16 bytes of secret key. The 16-byte secret key is stored in flash. Even with careful choices of block cipher parameters, there exist practical challenges in implementing RC5 on such a resource-constrained platform. Because of the extremely limited RAM memory (256 bytes), minimizing stack size is crucial. For instance, our first implementation resulted in an immediate stack overflow upon running the key scheduler. Since RC5 requires expanded key table of size $2(r+1)$ words, where r is the number of rounds, careful memory planning is required to reduce any unnecessary memory writes. The current implementation fully utilizes all of its RAM memory so that expanded key table ($2(r+1)*4$ bytes = 104 bytes) is kept in RAM along with just enough room for the stack. One of the approaches to remedy possible stack overflow would be to use a precomputed expanded key table. It can be precomputed and stored in ROM or flash as long as the secret key remains the same without hurting the performance since reading from flash or ROM takes the same amount of power and cycles as reading from RAM. In our implementation, we compute the extended key table once in every hardware reset. In other words, the key table is computed during the first active cycle and kept in RAM unless WISP reaches a brownout voltage. This decision is based on the assumption that it is unlikely for the secret key to be changed while an RFID reader is polling.

Although WISP can be running at 3MHz at 1.8 V, our data comes from running WISP at 3MHz at 3.0 V at the cost of reduced operating range to ensure reliable flash writes. Since many applications with security requirements assume a relatively short read range during the reader-tag communication, we believe that this is a reasonable demonstration of present capabilities, which may be enhanced for greater computation and range with newer, more efficient microcontrollers. There is already a newly released MSP430 that allows internal flash writes with only 2.2 V [4]. A more recent version of WISP not used in this paper has an external serial flash that operates at 1.8 V, allowing flash writes at much longer ranges. We expect the range to continue to increase as the energy-efficient performance of microelectronics continues to improve. Here, we aim to show any patterns or characteristics involved rather than providing exact figures.

All three major functions of RC5 — `setupKey()`, `encrypt()`, and `decrypt()` — have been implemented and measured on WISP. `setupKey()` is executed once after hardware reset, and the 64-bit ID value is encrypted or decrypted once every duty cycle (e.g., during “Generate Packet” stage in Fig. 2).

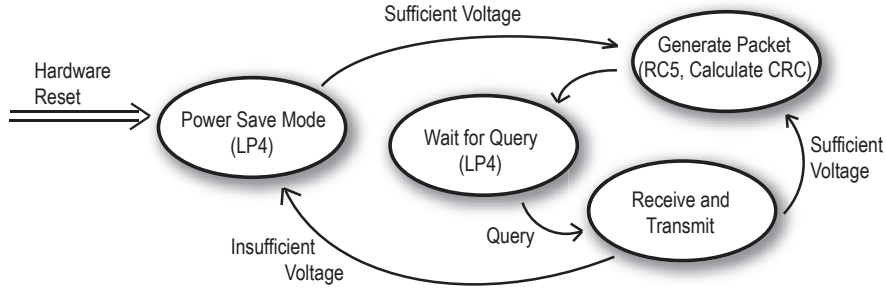


Fig. 2. WISP lifecycle

The duty cycling is enabled by a hardware voltage supervisor. This supervisor creates software interrupts by toggling a microcontroller input high to indicate sufficient voltage for operation. When the supervisor detects a voltage of 3.3V or greater, the microcontroller is enabled and a computation is initiated. At the end of the computation, if the voltage is between 2.7V and 3.3V the microcontroller enters very low power (RAM retention only) mode. While in this low power mode, the harvester provides power to re-charge the supply capacitor back to 3.3V for the next computation. Below 2.7V, a hardware reset is generated by an insufficient voltage indicator output on the voltage regulator.

Fig. 3 depicts a typical lifecycle of WISP with RC5 encryption enabled and voltage level patterns throughout the WISP duty cycle after a hardware reset, captured by an oscilloscope. The timeline shows that WISP spends most of its time in Power Save Mode(LP4), approximately 90-95% of its duty cycle. WISP is in active mode only during “Generate Packet” and “Transmit and Receive” stages.

Table 2 presents the execution time for three operations of RC5 on WISP tags operating at a distance of 1 foot (0.3 m). The time measurements are the mean of ten separate trials of execution after a hardware reset. The RFID reader constantly queries the WISP tag (about 15 polls per second) until the first response is received. Beyond the reader-tag distance of 2 feet (0.6 m), WISP generally does not function except for a few sporadic reads. However, within the range of 2 feet, we observed no major difference in execution time.

| RC5-32/12/16 function | Execution time (msec) |
|-------------------------|-----------------------|
| <code>setupKey()</code> | 7.93 |
| <code>encrypt()</code> | 1.43 |
| <code>decrypt()</code> | 1.39 |

Table 2. Execution time for operations in RC5 with 32-bit words, 12 rounds, and a 16-byte secret key on a WISP tag. The WISP received uninterrupted power. Encryption happens on 64-bit messages.

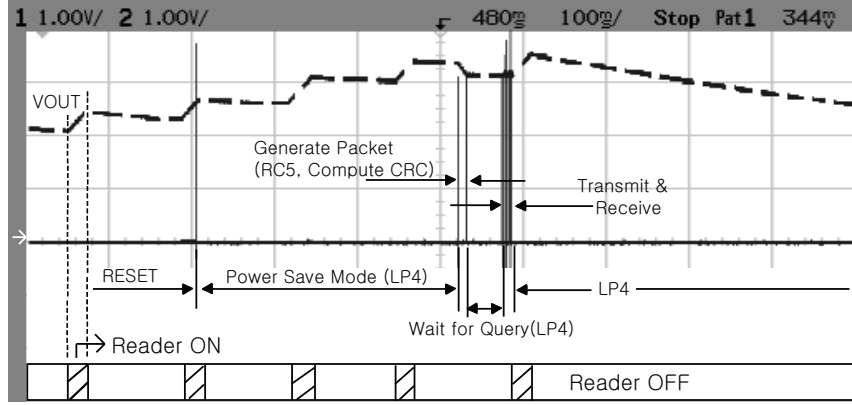


Fig. 3. An annotated scope trace of WISP execution and power consumption. The dotted line measures the supply voltage VOUT in 1V increments on the y-axis. While the Reader is ON (i.e., sending queries), the WISP’s voltage level stairs up. The ramp up from 0 V is omitted on the left. The voltage supervisor wakes up the WISP from LP4 when the voltage level exceeds 3.3 V, and WISP begins its computation (“Generate Packet”). When the reader receives the WISP’s response, the reader stops sending queries, therefore cutting off RF-power to the WISP. This is observed as a gradual decline of voltage at the right side of the figure. The WISP first enters LP4 and then resets as the voltage level falls below the minimum operating voltage. The total latency from 0 V until the end of RF response transmission in this case is approximately two seconds.

It is not apparent from the displayed data, but WISP with RC5 enabled does not greatly affect the average tag response time. At a distance of 1 foot, the difference in the average tag response time (e.g., time between when the first reader command is sent and the corresponding response is received) of WISP with and without RC5 is negligible (average of 462.6ms).

4.2 Estimates of Maximal Computation and Measurements of the Effect of Flash Writes on Computation

Fig. 4 provides an estimate of available microcontroller instructions versus wireless distance. The model used to generate this plot is based on experimental data of WISP performance, on the Friis transmission equation, and on published microcontroller power consumption specifications.

The Friis Transmission equation provides an estimate of power received as a function of distance from the transmitter, transmit power, antenna gains, and wavelength:

$$P_R = P_T - 20 \log \left(\frac{4\pi d}{\lambda} \right) + G_T + G_R \quad (1)$$

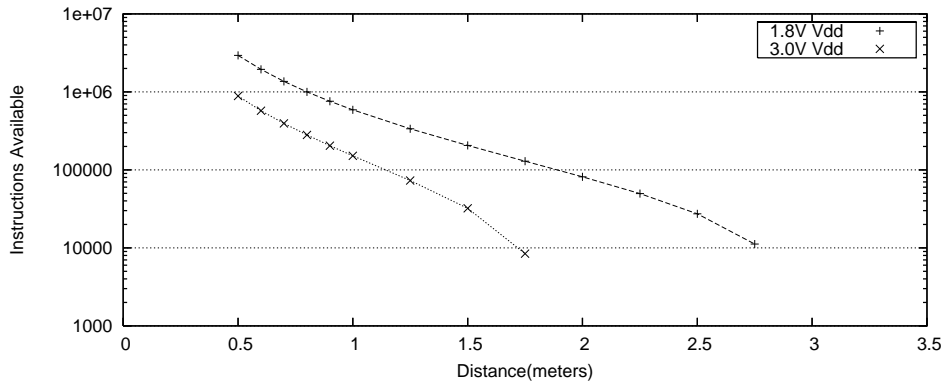


Fig. 4. This estimate shows the computation available in one active cycle, where the power harvester charges a 10 uF capacitor to its peak value and then the microcontroller enters active mode, discharging the capacitor down to the minimum operating voltage. The Instructions Available line ends abruptly at the distance where the harvester is no longer able to supply the microcontroller’s minimum operating voltage.

The RFID reader power $P_T = 30dBm$; the receive and transmit antenna gains are given by $G_R = 2dBi$ and $G_T = 6dBi$ respectively. The wavelength $\lambda = 0.33m$ at 915 MHz. Using the Friis equation with these parameters, we can find the expected power available as a function of distance.

In [23], experimental data of output voltage as a function of input power for the WISP power harvester is given. Thus this data, together with the Friis equation, allows us to predict the WISP’s output voltage V_{rec} as a function of distance d . This data captures the finite efficiency of the WISP harvester. An ideal, lossless harvester could produce any desired voltage if it were able to accumulate energy long enough. Actual WISP devices reach a steady state voltage, at which point power lost and power harvested balance.

The TI-MSP430 data sheet specifies its current consumption for given voltage and operating frequencies. This allows us to calculate its energy per instruction: 281 pJ per instruction at $V_{dd} = 1.8$ V or 900 pJ per instruction at $V_{dd} = 3.0$ V.

Of the energy stored in the WISP’s storage capacitor, only the voltage above the microcontroller’s minimum operating voltage is usable. If the capacitor is initially charged to V_{rec} and its minimum operating voltage is V_{dd} , then $\frac{1}{2}CV_{rec}^2 - \frac{1}{2}CV_{dd}^2$ Joules are available to run the computation. Using the previously calculated dependence of V_{rec} on distance, we find energy available as a function of distance. Dividing by energy per instruction provides number of instructions as a function of distance.

Our actual measurements in Fig. 5 show the relationship between the reader-to-tag distance and number of cycles available in one duty cycle with varying workloads of flash writes to the MSP430 information memory. While Fig. 4 provides the estimated upper bound on how much computation we can afford on WISP, Fig. 5 presents how many cycles are available after flash writes of

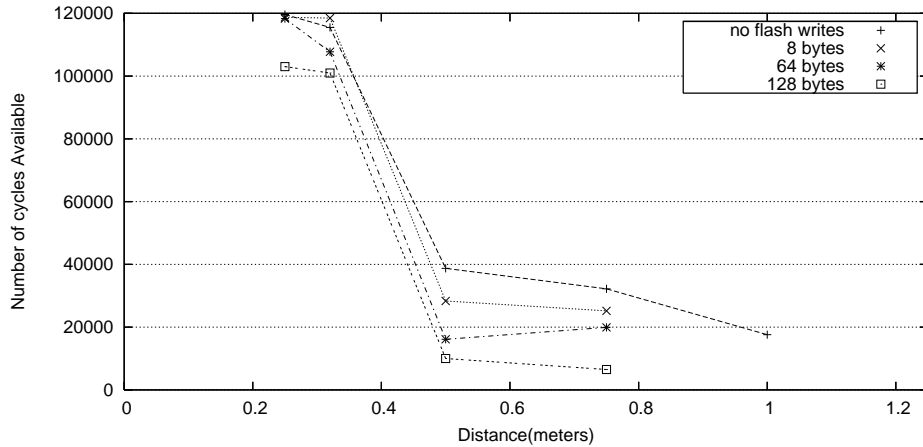


Fig. 5. Preliminary measurements of the number of cycles available after writing to flash — computed after a minimal set of operations during the “Generate Packet” stage and without tag response over RF. Each data point represents a mean of ten samples.

different sizes. For consistency, the experiment involves two steps, initialization and measurement. During the initialization, WISP’s flash memory is checked if its state is erased or not. The flash is wiped out (writing ‘1’ in all bits) before each measurement is taken. During the measurement phase, WISP consumes power writing to flash then raises a bit in a loop until WISP completely runs out of power (brownout voltage of 1.5V). These bit raises appear as a pulse on the oscilloscope. Since raising a bit requires 4 cycles, the number of bits are counted to compute available number of cycles after writing different number of bytes to the flash. Note that a minimal set of operations are done during the “Generate Packet” stage in this experiment (e.g., preloading the 64-bit ID into an array), and there is no RC5 computation or CRC computation.

We observed that the number of cycles available does not deviate significantly from each other at short read ranges. At the medium range of 0.5 m, the power consumption of a flash write causes a significant decrease in available computation. Beyond the distance of 0.75 m, WISP is not able to reach the minimum operating voltage of 2.7 V for flash writes, therefore the plot lines for flash writes end abruptly. These measurements are preliminary, and further research is necessary to determine a more precise relationship between flash writes, erase segment sizes, and computation. However, one conclusion is that unnecessary writes to flash at mid-range distances will significantly reduce available computation.

5 Conclusions

Our preliminary experimental data shows that UHF RFID tags with cryptographic capabilities are no longer infeasible. Although our experimental platform

exceeds the current EPC UHF Class 1 tags in terms of computing power and storage, we believe that the device is comparable and a good representation of the future of UHF RFID tags. We believe that the trend in microelectronics will continue to bring power-efficient and cost-effective microcontrollers capable of more sophisticated computation. We hope that our work will encourage further research to determine the feasibility of maximal-strength cryptography with actual power measurements of RF-powered UHF RFID tags — an beyond algorithmic estimates of space and running times.

Acknowledgments

We thank Dan Holcomb and Thomas Heydt-Benjamin for their assistance in debugging experimental setups and for giving critical feedback; Salma Mirza for her help in conducting experiments with WISP; and Peter Desnoyers and Gaurav Mathur for advice on the MSP430. This material is based upon work supported by the National Science Foundation under Grant No. 0627529.

References

1. Aigner, M.: DemoTag (2006) Last viewed May 16, 2007 http://www.iaik.tugraz.at/research/vlsi/02_products/05_rfid_demotag/.
2. Aigner, M., Feldhofer, M.: Secure symmetric authentication for RFID tags. In: Telecommunication and Mobile Computing – TCMC 2005, Graz, Austria (2005)
3. Bannister, K.: Cytheric:master's. WWW (2007) Last viewed May 12, 2007 <http://cytheric.net/masters/comm-test.shtml>.
4. Buccini, M.: New MSP430 family from Texas Instruments runs at 12 RISC MIPS and standby of 1uA. WWW (2004) Last viewed May 14, 2007 http://www.microcontroller.com/news/ti_msp4302xx.asp.
5. Calmels, B., Canard, S., Girault, M., Sibert, H.: Low-cost cryptography for privacy in RFID systems. In Domingo-Ferrer, J., Posegga, J., Schreckling, D., eds.: International Conference on Smart Card Research and Advanced Applications – CARDIS. LNCS, Tarragona, Spain, IFIP, Springer-Verlag (2006)
6. Cui, Y., Kobara, K., Matsuura, K., Imai, H.: Lightweight asymmetric privacy-preserving authentication protocols secure against active attack. In: International Workshop on Pervasive Computing and Communication Security – PerSec 2007, New York, USA, IEEE (2007) 223–228
7. Defend, B., Fu, K., Juels, A.: Cryptanalysis of two lightweight RFID authentication schemes. In: International Workshop on Pervasive Computing and Communication Security – PerSec 2007, New York, USA, IEEE (2007) 211–216
8. Israsena, P.: Securing ubiquitous and low-cost RFID using tiny encryption algorithm. In: International Symposium on Wireless Pervasive Computing, Phuket, Thailand, IEEE (2006)
9. Juels, A.: Minimalist cryptography for low-cost RFID tags. In Blundo, C., Cimato, S., eds.: The Fourth International Conference on Security in Communication Networks – SCN 2004. Volume 3352 of LNCS., Springer-Verlag (2004) 149–164
10. Karlof, C., Sastry, N., Wagner, D.: TinySec: A link layer security architecture for wireless sensor networks. In: Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004). (2004) 162–175

11. Kwon, D., Han, D., Lee, J., Yeom, Y.: Vulnerability of an RFID authentication protocol proposed at SecUbiq 2005. In: International Workshop on Security in Ubiquitous Computing Systems – Secubiq 2006. LNCS, Seoul, Korea, Springer-Verlag (2006)
12. Li, T., Wang, G.: Security analysis of two ultra-lightweight RFID authentication protocols. In: IFIP SEC 2007, Sandton, Gauteng, South Africa, IFIP (2007)
13. Microelectronic, E.: EM 4035 datasheet (2006)
14. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J., Ribagorda, A.: LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In: Workshop on RFID Security 2006(RFIDSec 06), Graz, Austria, Ecrypt (2006)
15. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J., Ribagorda, A.: M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In: International Conference on Ubiquitous Intelligence and Computing – UIC06. Volume 4159 of LNCS., Springer-Verlag (2006) 912–923
16. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: EMAP: An efficient mutual authentication protocol for low-cost RFID tags. In: OTM Federated Conferences and Workshop: IS Workshop – IS'06. Volume 4277 of LNCS., Springer-Verlag (2006) 352–361
17. Perrig, A., Szewczyk, R., Wen, V., Culler, D.E., Tygar, J.D.: SPINS: security protocols for sensor networks. In: Mobile Computing and Networking. (2001) 189–199
18. Piramuthu, S.: HB and related lightweight authentication protocols for secure RFID tag/reader authentication. In: Collaborative Electronic Commerce Technology and Research – COLLECTeR 2006, Basel, Switzerland (2006)
19. Poschmann, A., Leander, G., Schramm, K., Paar, C.: DESL: An efficient block cipher for lightweight cryptosystems. In: Workshop on RFID Security 2006(RFIDSec 06), Graz, Austria, Ecrypt (2006)
20. Ranasinghe, D.C., Lim, D., Cole, P.H., Devadas, S.: White paper: A low cost solution to authentication in passive RFID systems. Technical Report WP-HARDWARE-029, Auto-ID Labs, The University of Adelaide, Adelaide, Australia (2006)
21. Rieback, M., Gaydadjiev, G., Crispo, B., Hofman, R., Tanenbaum, A.: A platform for rfid security and privacy administration. In: USENIX/SAGE Large Installation System Administration conference – LISA'06, Washington DC, USA (2006)
22. Rivest, R.: The RC5 encryption algorithm. In Preneel, B., ed.: FSE. Volume 1008 of LNCS., Springer (1995) <http://theory.lcs.mit.edu/~rivest/Rivest-rc5rev.pdf>.
23. Sample, A.P., Yeager, D.J., Powledge, P.S., Smith, J.R.: Design of a passively-powered, programmable platform for UHF RFID systems. In: IEEE International Conference on RFID 2007. (2007)
24. Smith, J.R., Sample, A.P., Powledge, P.S., Roy, S., Mamishev, A.: A wirelessly-powered platform for sensing and computation. In: 8th International Conference on Ubiquitous Computing (UbiComp 2006), Orange Country, CA, USA (2006) 495–506
25. Spotlight, R.: 5 things must know for RFID starters. WWW (2006) Blog entry, Last viewed May 15, 2007 <http://www.innovex-one.com/blogs/>.
26. ThomasNet: Short range UHF EPC tag is designed for item-level tagging. WWW (2007) Last viewed May 12, 2007 <http://news.thomasnet.com/fullstory/482149>.
27. Westhues, J.: Proxmark3. WWW (2007) Last viewed May 16, 2007 <http://cq.cx/proxmark3.pl>.