Accelerometers and Randomness: Perfect Together

Jonathan Voris
Polytechnic Institute of NYU
jvoris@isis.poly.edu

Nitesh Saxena
Polytechnic Institute of NYU
nsaxena@poly.edu

Tzipora Halevi
Polytechnic Institute of NYU
thalev01@students.poly.edu

ABSTRACT

Accelerometers are versatile sensors that are nearly ubiquitous. They are available on a wide variety of devices and are particularly common on those that are mobile or have wireless capabilities. Accelerometers are applicable in a number of settings and circumstances, including important security and privacy domains. In this paper, we investigate the use of accelerometers for the purpose of *true random number generation*. As our first contribution, we discover that an accelerometer possesses two unique and appealing properties when used as an entropy source. First, contrary to intuition, an accelerometer can derive sufficient entropy even when it is stationary (i.e., not subject to perceivable acceleration). Next, and more importantly, the entropy of a stationary accelerometer can not be reduced in the presence of a variety of environmental variations or even under adversarial manipulations. This means that, unlike other sensors, accelerometers are resistant to changing environments, benign or otherwise. To support this claim, we develop a thorough experimental adversarial model for accelerometers that supply a system with entropy. To the authors' knowledge, this is the first real world model in the context of entropy collection.

As our second contribution, we demonstrate the validity of accelerometer based random number generation on an RFID tag, which is a highly resource constrained device. We present the design and implementation of our method on an Intel WISP tag and conduct several novel experiments to evaluate its feasibility. Our results indicate that a high quality 128-bit random number can be extracted using an accelerometer in about 1.5 seconds even when the sensor is in a stationary state. To our knowledge, this is the first random number generation technique that is known to be viable for RFID devices based on general-purpose hardware.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Miscellaneous

General Terms

Security

WiSec'11, June14–17, 2011, Hamburg, Germany.

Keywords

Random number generation, sensors, ubiquitous computing, computational RFID

1. INTRODUCTION

In this paper, we consider the difficult problem of *true random number generation* (RNG). RNG is a fundamental component of several cryptographic and security primitives, such as key generation, strong password generation, encryption, and authentication. It is also a necessary building block of randomized algorithms used in other areas of computer science. Our focus is specifically on hardware RNG rather than pseudo RNG which does not draw randomness from any external properties or occurrences. When constructing a random number generator, the most critical design choice is deciding which type of hardware, input interface, or sensor to use. Traditional desktop computers have many interfaces available to them from which they may draw entropy. Each of these interfaces comes with its own set of weaknesses, however. As an example, sensors such as microphones and wireless interfaces draw entropy from sources that are susceptible to environmental variations or are easy for a malicious entity to manipulate or monitor.

The problem of finding a satisfactory source of entropy is exacerbated by the resource constraints imposed by inexpensive devices such as Radio Frequency Identification (RFID) tags. The manufacturers of these wireless appliances often can not afford to include any hardware that serves a unary purpose such as the collection of entropy. As a result, they are forced to rely on whatever forms of input and sensory data collection are already presently available. The low memory, power, and computational abilities of RFID enabled devices further complicate the RNG endeavor. For an RNG solution to be applicable to RFID tags it must not consume much power, be highly computationally efficient, and require little storage space.

Given their utility, accelerometers are becoming increasingly ubiquitous, especially on mobile and wireless devices. Accelerometers are inexpensive, costing less than $1 [1], and can be added to devices which typically do not have them at little extra cost. For example, one can be added to a desktop or laptop computer using a USB dongle. They are also available on RFID tags [33, 37] and have already been utilized for RFID security and privacy primitives (e.g., see the Secret Handshakes work on context recognition [12]). Given these appealing features and multiple use cases, we set out to investigate whether accelerometers can be used as viable sources of randomness, and to determine how their entropy collection performance and capabilities compares to existing solutions.

An important metric on which an entropy source must be judged is its sensitivity. A sensor must be capable of picking up more detailed information about its environment than an adversary can

detect. If this is not the case, then it would be trivial for such an adversary to simply monitor the underlying physical phenomenon in order to predict the generator's output. We discover that accelerometers perform better than other types of sensors in this regard. They are more sensitive than intuitively expected, being capable of picking up even minute vibrations from afar. These devices are so perceptive that models in laptops and USB sticks have been repurposed by Cochran et al. to monitor early indications of earthquakes [9].

As our results demonstrate, *even stationary* accelerometers provide a satisfactory amount of entropy. That is, they are capable of providing sufficient randomness to enable the efficient generation of random numbers on computational RFID tags (such as Intel's WISP tags [33, 37]) and other computing devices. Despite this high sensitivity threshold, we find that accelerometers are resistant to several environmental changes and different types of manipulations by adversarial parties. Besides those that involve tampering with the sensor or the use of specialized equipment such as a centrifuge or vibration isolator, any benign changes or adversarial manipulations only increase the amount of entropy that is available to an accelerometer. These two key insights make accelerometers a unique RNG solution that is easily deployable on many platforms.

Overview of Contributions and Paper Outline.

In this paper, we make the following technical contributions. We present two unique and appealing properties of an accelerometer when used as an entropy source. *First*, contrary to intuition, an accelerometer can derive sufficient entropy even when it is stationary, i.e., not subject to perceivable acceleration. *Second*, and more importantly, accelerometers are resistant to a variety of environmental variations and even to adversarial manipulation. To substantiate these claims, we develop an adversarial model for an accelerometer being used as an entropy source (*Section 3*). Our results demonstrate that most benign or adversarial changes that an accelerometer can be subject to will *increase* the entropy it provides. The best approach an attacker could take to interfering with the amount of accelerometer generated entropy would be to place the accelerometer equipped device in as stable an environment as possible.

As our second contribution, we demonstrate the practicality of our proposal. We design and implement an accelerometer based random number generator on an RFID tag (WISP), which is a highly constrained device (*Section 4*). We also report on the detailed results regarding the novel experiments we performed to measure the amount of randomness one can expect to derive from an accelerometer while it is undergoing a variety of motions and circumstances. Our results indicate that a high quality 128-bit random number can be extracted using an accelerometer on a WISP tag in about 1.5 seconds in a stationary state and much faster when an accelerometer equipped device is used and carried during daily activities.

Additionally, we show that accelerometer based RNG compares favorably to existing RNG solutions in terms of many metrics. Accelerometers are universal and capable of functioning irrespective of how they are stowed since they function when placed inside of other objects. Due to these features, accelerometer based RNG can work on routers and servers that lack traditional interfaces, and on RFID enabled devices that are often kept inside wallets or purses. We corroborate our results with a thorough comparison with related work on alternative sources of randomness (*Section 6*).

2. BACKGROUND AND PRELIMINARIES

2.1 Random Number Generation Theory

Cryptographic applications demand "strongly" uniform numbers. The bits of the number must be independent and uniformly distributed, or as close to this as attainable. If this type of random value was naturally occurring, utilizing it would be a relatively simple matter of recording it and handing it to the cryptographic application. Unfortunately, this type of strong randomness is unlikely to be available in practice. While the naturally occurring phenomena that sensors capture are unpredictable, they necessarily contain some bias rather than being distributed uniformly.

Extraction functions have been created to address the above problem. An extractor is a function that takes a string of unpredictable but biased, or "weakly" random, bits as input and returns a string of close to uniform, or "strongly" random, bits as output. One example of such an extractor is the "independent sources" extraction of Barak, Impagliazzo, and Wigderson [5], which simply works by multiplying two independent values and adding the result to a third in a recursive fashion. Along the same lines, a second type of extractor was described by Barak, Shaltiel, and Tromer [6]. This extraction technique utilizes a Toeplitz matrix as a seed, which is multiplied against the column matrix containing the input to the hash function. Both of these extractors produce streams of output that are provably close to uniform when provided with inputs which possess sufficiently high min-entropy. Min-entropy, a mathematical property of a distribution, is defined as follows:

DEFINITION 1. *The min-entropy of a given distribution X on $\{0,1\}^n$ is:*

$$min\text{-}entropy(X) = \min_{x \in \{0,1\}^n} \log_2 \frac{1}{Pr[X = x]}$$

In words, the min-entropy of a distribution is equal to the probability of the most likely element in X being drawn from X. Phrased somewhat differently, if a distribution X has a min-entropy of k, the likelihood of drawing any single element x from X does not exceed $1/2^k$ for all $x \in X$.

Min-entropy is an important measurement of a distribution because it captures the amount of randomness a distribution is capable of supporting. Despite the fact that elements of X are n bits in length, due to the bias of the distribution, X may not contain enough entropy to actually support the extraction of n unbiased bits. Only k "strongly" random bits can be derived from a distribution that has a min-entropy of k regardless of the distribution's element length n.

With the concept of min-entropy established, the definition of an extraction function can be expanded in more detail.

DEFINITION 2. *A (k, ϵ)-extractor is a function of the form:*

$$F : \{0,1\}^n * \{0,1\}^d \rightarrow \{0,1\}^m$$

where, for every distribution X over $\{0,1\}^n$ with min-entropy $\geq k$, the output of $F(X, s)$ is statistically ϵ-close to the uniform distribution over $\{0,1\}^m$ when s is chosen uniformly at random, $s \in_R \{0,1\}^d$.

Thus, a (k, ϵ)-extractor is nothing more than a function that accepts n bits of input with min-entropy k and a d bit seed and outputs m bit long values that are nearly uniform. Here, "strongly" random numbers have been described as being "ϵ-close" to uniform.

2.2 WISP Tags

To investigate how to meet the RNG needs discussed above, we mainly utilized a special type of RFID tag designed by Intel Research known as a Wireless Identification and Sensing Platform (WISP) [33, 37]. WISPs are passively-powered RFID tags that are compliant with the Electronic Product Code (EPC) protocol. Specifically, we utilized the 4.1 version of the WISP hardware,

which partially implements Class 1 Generation 2 of the EPC standard. Where the WISP differs from standard tags, however, is in its inclusion of an onboard Texas Instruments MSP430F2132 microcontroller and sensors such as the ADXL330 three-axis $\pm 3g$ accelerometer. This 16-bit MCU features an 8 MHz clock rate, 8 kilobytes of flash memory, and 512 bytes of RAM. WISPs are the first programmable passive RFID devices. They have seen use in studies on a variety of topics, from the energy harvesting experiments [24, 23] to monitoring animal behavior [21, 35]. Unlike standard RFID tags, which are fixed function and state machine based, the flexibility of the WISP allowed us to implement novel security solutions on a live, passive RFID device. The mass manufacturing cost of a WISP tag is expected to be close to $1 [8].

3. ADVERSARIAL MODELING

A prerequisite to building a secure RNG system is to understand how the underlying entropy source behaves in the presence of benign or malicious changes in the context the system is deployed in. In particular, it is important to determine whether or not the min-entropy of the output distribution of the source is affected under different operating conditions, and if so, to what level. If the min-entropy can be reduced to less than a predetermined value (or, in the worst case, brought down to zero), then the extraction function will not be able to guarantee a near uniform distribution for the numbers generated, thus undermining the system's security. For example, the distribution of a microphone or other audio sensor's output will be influenced by the sound produced by users in close proximity, among other environmental factors. Thus, if an adversary can supply a constant audio input or loud noise to the microphone, the system can be forced into a zero entropy state.

In this paper, we develop an experimental adversarial model for an accelerometer being used as an entropy source. In order to achieve this, we analyze what factors, malicious or otherwise, affect the output – and therefore the min-entropy – of the accelerometer, and to what extent this occurs.

The values that are output by an accelerometer are a function of the following variables: *acceleration*, *noise*, *sampling rate* and *temperature*. Our model is driven by the question: *can an adversary who tries to manipulate these variables reduce the min-entropy to a level lower than an expected value?*. As our test sensor, we use an onboard WISP accelerometer, model ADXL330 [2], which is also commonly used on other low-end devices. We also perform some experiments with a mobile phone accelerometer (specifically model LIS302DL) which can be found on Nokia N97 cell phones [28, 38]. We analyze different accelerometer input variables and their affect on min-entropy generation below.

1. Acceleration: Clearly, an accelerometer's output depends on what external acceleration the sensor is subject to. Acceleration is defined as a change in velocity. The output of an accelerometer typically varies linearly with acceleration, as is the case for the ADXL330 accelerometer. Our experiments for common and benign movement and acceleration scenarios also confirm that min-entropy increases with the amount of motion applied to an accelerometer equipped device, and that, out of all potential motions, stationary state samples yield the lowest min-entropy. More details on this are provided in Sections 4.1 and 4.6.

We further test some specialized scenarios where an RFID tag may experience either a very low or a constant acceleration that may affect the amount of min-entropy its onboard accelerometer produces (refer to Section 4.3). Both sets of experiments attempt to explore the possibility for an intruder to significantly reduce the amount of min-entropy that is being derived. Our results show that the min-entropy level can not be lowered considerably.

2. Noise: Another important parameter that impacts the randomness of accelerometer output is the sensor's noise. This includes intrinsic noise generated from within the accelerometer circuitry as well as that induced by the environment (typically referred to as seismic noise). As specified in [2], this noise follows a Gaussian distribution. This was also confirmed by means of a set of our stationary state samples (Figure 1). Note that the peak of this distribution is what corresponds to the min-entropy; the flatter the curve, the higher the min-entropy.

We note that an accelerometer's noise is random and its overall level can only be lowered by reducing the bandwidth of the accelerometer, which would in turn increase the resolution of the sensor. This can only be performed by changing the capacitances on the accelerometer circuitry [2], which requires physical access to the device. Therefore, it would not be possible for an adversary to manipulate the amount of noise present in order to cull the min-entropy. The WISP schematic depicts the default bandwidth to be 50 Hz, which corresponds to a capacitance value of 0.1 μF for each of the three axes [13]. All of our experiments reported in this paper were performed at this default setting.

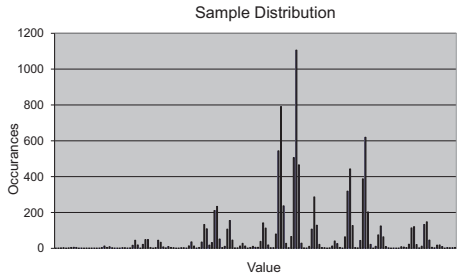

Figure 1: A Stationary State Sample Distribution taken from the WISP's Accelerometer

3. Sampling Rate: The sampling rate of an accelerometer's analog-to-digital converter (ADC) is an important measure of the sensor's output. Our experiments indicate that the rate at which an acceleration sensor is sampled does not have a significant effect on the min-entropy of its output (see Section 4.4 for details). Notice that in certain applications, such as a passive RFID system, a malicious reader can control the sampling rate in an attempt to undermine the level of randomness that is produced from a tag's accelerometer.

4. Temperature: We review the effect of temperature on an accelerometer's output and on the min-entropy level. According to the ADXL330 specification, the effect of temperature on the accelerometer's sensitivity is very low. It is only $\pm 0.015\%$ per $^\circ$C , or 1% for 70°C. Further, the bias change is ± 1 mg/$^\circ$C with a maximum of 70 mg for 70°C. In addition, our tests confirm that temperature does not have a considerable impact on the level of accelerometer min-entropy (see Section 4.5). The default temperature for experiments was the room temperature in our lab.

Our experiments show that an adversary who tries to manipulate an accelerometer's input can not reduce the min-entropy level. No matter how these inputs are modified, it does not seem possible to reduce the min-entropy found in the sensor's output beneath what is present in a stationary sample taken from the sensor. In other words, our model establishes a safe rough lower-bound for the min-entropy of accelerometer readings. We note that this bound is an essential parameter for the extraction approach and model that is presented in [6], which is what we employ for our extraction needs.

To summarize, according to our experimental model, accelerometers turn out to be resilient to adversarial control. The best ap-

proach an attacker could take in terms of interfering with the amount of min-entropy generated by one of these sensors would be to place the accelerometer equipped device in as stable an environment as possible, as anything else will only serve to increase the min-entropy of its readings rather than reduce it.

It may be possible to perform a more sophisticated form of attack to reduce the amount of randomness in an accelerometer's output by ensuring that the device is constantly undergoing precisely the same or a very high amount of force. For example, an adversary could place an RFID tag in a centrifuge that spins the tag at a very high speed, pinning the accelerometer readings to the same maximum value. On the other extreme, a perfect vibration shield could be used to completely cut a device's accelerometer off from the external phenomena which it draws its entropy from. An intermediate possibility that we experimented with in Section 4.7 involves inducing a resonance effect [3]. If an accelerometer is exposed to a sustained force at a particular frequency, known as its "resonant frequency", the amplitude of the output signal will grow significantly, causing the signal to "clip" or saturate. This will cause the accelerometer to constantly output same maximum output value, yielding zero min-entropy. We could not, however, successfully exploit this phenomenon with the LIS302DL accelerometer by using its specified resonance frequency of 2000 Hz [22]. (This experiment is further detailed in Section 4.7).

While these types of attack may succeed, they are not very practical. First, to perform them, physical access to the device for an extended period of time is needed, at which point an adversary could instead physically decompose the tag to compromise the integrity of the entropy collection process as well as any secrets that are stored on the device. Second, these attacks may be easily detected. Third, it might be hard to design a true vibration shield that can shield an accelerometer from all vibrations.

4. ACCELEROMETER RNG ON RFID

Like most devices, RFID tags are in need of RNG. One motivating example is that random values are a prerequisite for executing RFID tag-to-reader authentication protocols, such as HB+ and HB# [25, 14]. Privacy-preserving authentication protocols also require unpredictable numbers [29]. In this section, we discuss the design, implementation, and experimental analysis of an accelerometer based random number generator for RFID devices.

As discussed in Section 1, RNG is beyond the capacity of today's average RFID tag. As a result, alternative approaches to the creation of random values must be considered. One such proposal is proposed by Holcomb et al. [19, 20]. This technique utilizes onboard RAM as a source of true randomness. This technique is quite promising as any device, regardless of its constraints, will contain some amount of onboard memory from which randomness can be drawn.

Unfortunately, previous work has illustrated that practical considerations prevent the FERNS approach to random number generation from reaching its full theoretical potential [34]. Since FERNS relies on pre-existing memory circuitry as a source of entropy, it must compete with other system functionalities for use of this shared resource. Other code running on an RFID tag will necessarily be occupying the device's memory at any given point during execution. As such, the amount of uninitialized RAM available for utilization as a randomness generator may be restricted to a fraction of such a device's overall memory.

Furthermore, RAM is subject to a phenomenon known as *data remanence*. While it is still volatile in the traditional sense, due to properties of the underlying hardware such memory retains its contents while receiving power and for a duration of several seconds afterwards, as discussed by Skorobogatov and Halderman et al. [36, 17]. This means that after a portion of memory has been used for entropy collection once, it will require a relatively extended period of time without power before it can again be used in this capacity. In a usable RFID based security application which requires multiple random numbers this may lead to unacceptably high delays. As an alternative, we instead turn to entropy collection techniques which rely on onboard sensors. While not as general purpose as RAM, sensors have many uses outside of security and privacy applications. Note that not all sensors qualify to serve this purpose. RFID devices are often stowed inside other objects. For instance, access cards are often stored inside of a wallet or purse. This rules out the use of sensors such as microphones, cameras, or light sensors. See Section 6 for a more thorough sensor comparison. Accelerometers, on the other hand, appear to be a promising foundation for performing RNG on RFID tags.

4.1 Min-Entropy Estimation

To investigate the viability of generating cryptographic quality random numbers using accelerometer readings on mobile hardware, several experiments were performed. First, we needed to approximate the min-entropy of the accelerometer samples intended for extraction. Accelerometer samples were taken over a 10 minute interval while a variety of different movements were performed with the tags. In all cases, min-entropy was calculated by applying the following process. After collecting a sample of accelerometer readings, the number of occurrences of the most common value in the sample was counted. The probability of choosing this element is calculated by dividing this number by the total number of elements in the sample. The min-entropy of the sample distribution was computed by applying Definition 1 to this value.

Movement	Min-Entropy
Stationary #1	3.4
Stationary #2	3.6
Hand	10.8
Arc Swipe	11.3
Drop	9.1
Triangle	11.0
Alpha	11.0
Key Twist	11.7
Circle	11.4

Table 1: Min-entropy Estimates of Accelerometer Sample Distribution for 10 Minute Motion Samples

The sample with the least amount of motion involved was the stationary sample, where the WISP tag was simply left sitting on a desk. This test was meant to model a scenario where a tag is placed in front of an RFID reader's antenna without actually being held by a user. The hand test measured the min-entropy of the accelerometer readings while the WISP tag was held in the palm of a hand. This test was meant to model a scenario where a tag is presented in front of an RFID reader's antenna while being held as still as possible by a user. The arc swipe sample involved moving the WISP tag in an arc like half circle pattern from the middle left hand side of the reader's antenna, to the center top of the antenna, then to the middle right hand side of the antenna, and then back again. This test was meant to model a scenario where a tag is swiped in front of an RFID reader's antenna while being held by a user.

For the drop test, the WISP tag was repeatedly picked up and vertically dropped in front of the antenna. This test was meant to stimulate items being deposited in front of a RFID reader as they move down a conveyor belt in a factory or retail checkout. In

Movement	Sample Size	Min-Entropy
Overnight #1	1,231,095	3.5
Overnight #2	2,778,113	3.9

Table 2: Min-Entropy Estimates of Accelerometer Sample Distribution for Overnight Samples

the triangle test, the WISP tag was moved in a triangular pattern from the bottom left hand corner of the reader's antenna, to the top center of the reader's antenna, then to the bottom right hard corner, before being moved back to the bottom left. For the alpha sample, the tag was moved in a loop resembling a lower-case Greek letter alpha. Both the alpha and triangle tests were also meant to model a scenario where a tag is swiped in front of an RFID reader's antenna in a certain manner while being held by a user.

Instead of moving the tag parallel to the reader surface, for the key twist test, the tag was held in place relative to the antenna but spun in circles around its central axis, similar to the motion performed when a key is used to open a door. This test represents the motion underwent by an RFID tag embedded in a key while unlocking a door. Finally, the circle test saw the WISP tag moved roughly in a circle in front of the antenna, once again to model a scenario where a tag is swiped in front of an RFID reader's antenna in a certain manner while being held by a user. The arc swipe, triangle, alpha, key twist, and circle motions were first suggested in the study of Secret Handshakes [12] and were included to provide a basis for comparison with this work.

The results of these tests are given in Table 1. Out of all these patterns, the stationary option yielded the lowest min-entropy with a value of 3.4. To verify the accuracy of this result, a second 10 minute stationary sample was taken. The min-entropy of this sample was found to be slightly higher than the first, 3.6. Thus it was concluded that a min-entropy level of approximately 3.4 should be assumed for accelerometer outputs, since it is unknown how much motion, and therefore how much additional min-entropy, will be captured by the samples at any given time. Note that the min-entropy of a sample distribution captures an estimate of the amount of randomness that can be expected to be derived from a single sensor reading rather than the entire distribution sample. That is, a stationary accelerometer can support the creation of 3.4 random bits per one 30-bit accelerometer sample. Due to the limitations of our sample sizes, these values should be regarded as min-entropy lower bounds rather than definitive min-entropy estimations.

Our tests determined that the min-entropy of the RFID tagŠs accelerometer samples is at its lowest when the tag is still. This was further confirmed by a series of specialized experiments reported in Section 4.3 to 4.7). To further ensure an accurate estimate of the sensor value's min-entropy for this (worst-case) scenario, a sizeable sample was needed. To achieve this, a tag was programmed to transmit its raw accelerometer values upon receipt of a query from a reader. The reader was left to query the tag overnight twice. The results of these tests are given in Table 2. The 1,231,095 readings collected in the first sample yielded a min-entropy of 3.5 while the second batch's 2,778,113 readings had a min-entropy of 3.9. These values confirmed that the original min-entropy estimate was accurate and not due to a chance in the smaller sample.

4.2 Extraction

In order to produce a uniformly distributed random value, we utilized known extractor functions. However, since the extraction was to be implemented on a WISP tag, which has limited resources, special considerations were necessary. Extractor functions, reviewed previously in Section 2.1, were used to achieve this goal. Specifically, we used the independent sources extractor presented in [5] and the matrix extractor presented in [6], as described below. (Although we concentrate on generating a 128-bit random number, our approach presented in this section can be generalized to produce an arbitrarily long random number).

Chained Extraction: For efficiency purposes (due to the resource limitations), we decided to implement the extraction using a two-stage process, where the output of the independent sources extractor was fed into the matrix extraction function as input. The first (independent sources) extractor was utilized to create a compressed output and allowed us to minimize the amount of input for the second (matrix) extractor, which was then used to generate a 128-bit random output. The primary advantage of this approach includes reducing the input for the second extractor, which in turn significantly reduces the computation required. Another benefit of this approach is that it can be easily generalized to create random output that is longer than 128-bits (even on limited computation devices).

Extractor Function Details: As mentioned above, for the first stage in our extraction process, we implemented the independent extraction technique. We used as input the three axes of an accelerometer sample (which were 10-bits long each), resulting in a total 30-bit input which produced a 10-bit output. A core advantage of this extractor lies in its simplicity. Since it only involves one multiplication step and one addition step, it can be readily deployed on platforms that lack the computational resources. Unfortunately, the independent sources extractor can not be used on its own to craft 128-bits of randomness. Since each input to this function is only 10-bits in our case (i.e., we only had three axes of accelerometer output acting as three independent sources), it can only be used to generate a 10-bit long random number.

For the second stage, we applied the matrix extractor. For example, to produce a 128-bit random output corresponding to our stationary state samples, a 50 sample input to the extractor was used that had a min-entropy equivalent to about 198, which was necessary for the matrix extractor [6]. Since the extractor input consisted of the first stage output, this resulted in 500 bits input length.[1]

This entropy extraction technique is more flexible than others since it provides a method to control its input length, the size of its output, and how close to uniform its results will be [6]. Unfortunately, it has larger input requirements than its alternatives (when used alone), which makes the matrix extractor harder to use, as a single-stage process, on resource constrained devices.

Implementation Details: To implement the two-stage or double extractor on a WISP tag, several changes had to be made. These changes were necessary, in particular, for the matrix extractor to work within the constraints of the tag. Only the top row and left-most column of the Toeplitz matrix seed are permanently stored on the tag. When performing matrix multiplication operations, each row of the matrix was generated as needed from the seed and discarded afterwards in order to minimize the amount of memory needed to store the seed. Furthermore, all binary values are stored in byte arrays rather than arrays of boolean values. While this adds complexity to the manipulation of individual bits, it reduces the required storage space. In addition, rather than buffering accelerometer samples prior to applying the extraction, the matrix operations were done on a piecemeal, sample-by-sample basis, saving both memory as well as computation.

NIST Test Results: To confirm the randomness of our double extractor output, this approach was further implemented on a laptop computer and applied to each of the motion samples described in Section 4.1 as well as the two overnight samples which were taken. The movement samples were run through the National Institute of

[1]The extractor seed was 627-bit long.

Standards and Technology (NIST) "Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications" [32] both prior to and following extraction. The frequency, frequency within a block, cumulative sums, runs, longest-run-of-ones in a block, binary matrix rank, non-overlapping template matching, overlapping template matching, Maurer's "Universal Statistical," approximate entropy, serial, and linear complexity tests from the NIST suite were applied to the sample data[2]. The results of these tests are provided in Table 3.

Movement	% of NIST Tests Passed
Overnight #1	100.0%
Overnight #2	99.4%
Stationary #1	98.8%
Stationary #2	96.9%
Hand	98.8%
Arc Swipe	98.8%
Drop	97.5%
Triangle	93.8%
Alpha	98.1%
Key Twist	98.1%
Circle	97.5%

Table 3: NIST Test Suite Results for Double (Independent, then Matrix) Extracted 10 Minute Motion Samples and Overnight Samples

We find that the longer input samples passed a very high percentage of the test (with the overnight samples passing either all or 99.4% of the attempted tests). Since a relatively large number of samples are needed for proper statistical results, the smaller samples (such as the triangle data which only included 250 samples) returned lower results but still passed at least 93.8% of the tests. We therefore conclude that the NIST test indicate that our double extractor generated data with a sufficient level of randomness.

4.3 Effect of Vibration Shielding and Specialized Motion

We conducted several tests using commercial anti-vibration pads to garner insight into the effect of vibration shielding on the min-entropy of accelerometer readings. The pads used in our tests are rubber blocks that were originally designed to absorb distracting motion caused by large appliances such as washers and dryers. The intention behind these experiments was to isolate a WISP tag, and therefore its accelerometer, from external vibrations. We anticipated that this would prevent the sensor from picking up any external vibrations and that the min-entropy estimate of its readings would consequentially be lower.

The results of these tests, shown in Table 4 above, were surprising, however. The min-entropy reported by the WISP when the first sample was taken with it placed on the pad, 3.3, was only 0.1 lower than the baseline value taken with the tag placed directly on a desk. The second sample taken with a WISP on top of these pads revealed a slightly lower min-entropy value of 2.5. This is still within the range of usual values for the tag when it is at rest, however. The anti vibration pads thus had little impact on the min-entropy levels exhibited by the accelerometer readings. There are several possible explanations for this unexpected result. Since the pads were intended to dampen the impact of large vibrations from household appliances, perhaps they do not shield against the minute motion that the accelerometer we employ is capable of detecting.

[2]Using the default NIST test variables and parameters.

Test	Min-Entropy
WISP Stationary on Desk	3.4
WISP Stationary on Pad #1	3.3
WISP Stationary on Pad #2	2.5
WISP Stationary under Pad	3.0
WISP Stationary Pad Sandwich	2.0
WISP Dropped on Pad	7.4
WISP Slid Down Inclined Plane on Pad	6.2
N97 Stationary on Pad	1.4
N97 Stationary under Pad	1.4
N97 Stationary Pad Sandwich	1.3
WISP Salad Spinner	10.2

Table 4: Min-entropy Estimates of Accelerometer Sample Distribution for Shielding and Specialized Motion Tests

Alternatively, this could indicate that the bulk of the randomness output by the accelerometer comes from internal sources of noise rather than external vibrational motion. Finally, the accelerometer may have still been picking up vibration from the air above the tag rather from the surface and ground beneath it.

To refine these results, we took several samples with the WISP tag positioned differently relative to the anti vibration pads. To see if the device's accelerometer was being influenced by subtle motion from above, we "sandwiched" a tag between two pads. This resulted in a more dramatic min-entropy estimate decrease, as this value fell to 2.0. Yet, we were uncertain as to whether this effect was caused by the isolation of the tag from external movement or simply because the weight of the pad kept the WISP tag pinned down. To determine which was the case, we conducted a test where the tag was placed under a pad but did not have a pad underneath it as well. This resulted in a reading of 3.0, representing a higher amount of min-entropy. It therefore appears that the anti vibration pads did indeed shield the WISP tag from external motion, but this must be applied from all sides for the impact to be discernible.

To ensure that our results were generalizable to all devices and not simply limited to WISP tags or devices with an ADXL330 accelerometer model, we also ran tests with a Nokia N97 phone's LIS302DL accelerometer (a comparison between the min-entropy contained in the output of the LIS302DL and ADXL330 acceleration sensors is provided in Section 5.2, the former being lower). The results of this trial largely mirrored those performed with a WISP tag. Placing the phone under or on top of a pad diminished the min-entropy of the sensor's samples to identical values of 1.4, which represented the lower end of the phone's accelerometer while at rest. Placing the phone between two pads again had a larger impact, bringing the min-entropy level of the phone's accelerometer values down to 1.3.

The drop test listed in Table 3 was also reproduced, only in this variant the tag was dropped on to an anti-vibration pad. Although the tag was observed to bounce upon impact with this rubber surface, the results indicate that the pads did reduce the impact at the end of the fall somewhat, as the min-entropy estimate of the sample taken without padding, 9.1, was higher than the 7.4 obtained when the anti vibration pads were utilized. All these experiments indicate that commercial vibration pads are unlikely to shield a large fraction of movement from being read by an accelerometer. This substantially limits possibilities for adversarial action on an accelerometer based RNG system.

We also conducted an inclined plane test in order to study the performance of the tag under different constant force circumstances, as mentioned in Section 3. The goal was to have the device sustain constant accelerations in order to cause its accelerometer read-

ings to remain similar, thus lowering the min-entropy. A book was propped up adjacent to a rubber pad and a WISP tag was repeatedly slid down this surface. The positions of the book, rubber pad, and reader antenna were fixed from sample to sample. The amount of force applied to the tag's accelerometer was controlled by repeatedly picking the tag up, placing it at the top of the book, and allowing it to slide down along the book's front cover. One ten minute sample was taken. This motion sample yielded a min-entropy of 6.2, which is slightly beneath that of the free fall trials.

Finally, we took readings with a WISP tag placed inside of a swirling salad spinner. This kitchen tool was used as an impromptu budget friendly centrifuge. The purpose of this test was to provide constant acceleration by keeping the spinner running at a close to constant speed. The tag was taped to the inside of the moving portion of the spinner and read while the device was in motion. The spinner was regularly pumped to prevent the tag from slowing down during the test. As was the case for the inclined plane tests, the placement of the reader antenna and salad spinner was not altered during the tests. The variable element in this experiment was the location of the tag as it rotated along the inside walls of the salad spinner. A single ten minute trial was also conducted for this test. The resultant min-entropy estimate of the tag's accelerometer values was found to be 10.2. This is comparable to the results observed during the motion tests without specialized equipment rather than being indicative of any adversarial ability to reduce the randomness inherent in the accelerometer output.

Contrary to our intuition, the scenarios that were intended to feed constant force in to the accelerometer resulted in an increase in the min-entropy of its output. This seems to indicate that noise contributes significantly to the randomness contained within accelerometer samples. These experiments provide an initial estimation as to the degree of influence an adversary is capable of exerting over the output of an accelerometer. However, it must be noted that these results should not be taken as conclusive proof that a potential attacker would not be able to do better.

4.4 Effect of Sampling Rate and Method

Our experiments indicate that the rate at which an accelerometer is sampled does not have a significant impact on the min-entropy of its outputs. A function estimating the impact of the sampling speed is portrayed in Figure 2, which was moved to the Appendix. This figure implies that even at a sampling rate of infinity (i.e., when the time interval between successive reads is 0), the min-entropy of the sensor's output distribution would stand at around 3. The most feasible explanation for this behavior is that the entropy present in accelerometer samples comes mostly from noise. If the sampling rate used is smaller than the accelerometer's bandwidth, which should be the case to allow the ADC to work, then each reading is affected by a different noise level. This is because the state of the accelerometer always changes between each reading.

The next element that we looked at as a potential contributing factor to the accelerometer's min-entropy level was the sampling method employed. The WISP firmware has two separate techniques for taking readings from the acceleration sensor. One is a "quick" technique that does not allow the accelerometer to fully settle before taking readings, but instead takes a fast reading and then attempts to compensate for the error in the hasty sample. The other allows the accelerometer to settle completely instead. The "quick" reading technique is used by default, so all samples taken thus far have utilized this sampling technique. To see if the "slow" sampling technique resulted in a distribution with significantly different min-entropy from the default "quick" technique, an overnight test was conducted again, this time with the WISP firmware set to use the "slow" technique. The resultant min-entropy estimate of this sample was 3.4.

4.5 Effect of Temperature

We then set out to determine if temperature plays a role in determining the amount of randomness contained in these values. In order to see if temperature had any impact on the min-entropy of the accelerometer values whatsoever, we first used a blow dryer to cause a WISP tag to warm up and a freezer to cool it down. The tag was programmed to transmit the output of both its accelerometer and its internal thermometer to the RFID reader through its EPC ID. For the "hot" test, the WISP tag was placed in front of a polling reader's antenna while a blow dryer was aimed directly at the tag. The blow dryer was placed on the highest setting that we could use. For the "cold" test, the tag was placed in a plastic bag in the freezer section of a mini-refrigerator overnight. Immediately upon removing the tag, it was interrogated by an RFID reader. Both samples were taken for a period of 10 minutes as with the movement tests. The results of these tests are shown in Table 6 in the Appendix.

Since the min-entropy of the preliminary "hot" test sample was dramatically different from that of the stationary tag samples we previously encountered, we could not rule out the possibility that temperate did indeed have an impact on the randomness of the accelerometer samples. However, in both the cases of the "hot" and "cold" test, we were not able to exert as careful control over the temperate as we would have liked. This is because the temperature of the "cold" tag began to rise as soon as it was removed from the freezer, while the heightened min-entropy of the "hot" tag could have been caused either by the increase of temperature from the blow dryer or the buffeting of the air being blown at the tag by the blow dryer. Thus, as a follow up, we performed 3 additional temperature tests. In the first two, an electric heater was used in place of the blow dryer.

In one of these tests, the heater was set only to blow air on the tag, allowing us to isolate the effect of the force of air on the tag's accelerometer readings without simultaneously warming the tag. In the second, the heater was set to produce warmth in order to replicate the outcome of the blow dryer test with a different device. As a third test, to obtain a more stable cold temperature reading, the WISP tag was placed in a plastic bag and sealed in a thermos full of ice. After waiting several seconds to allow the tag's temperature to cool to that of the thermos, the thermos containing the tag was placed in front of the antenna of our RFID reader and queried for a 10 minute interval.

The min-entropy did not change much for the freezer test. This is because there was the least variation in its temperature out of all of the temperature tests performed. The heater and thermos tests each saw drastic changes in temperature, yet only saw modest increases in their min-entropy level. The fan test saw a substantial increase yet did not involve any temperature change at all, while the blow dryer test had the largest net gain of all the temperature tests performed. Since all tests in this group were performed with the WISP tags at rest, we can conclude that while temperature does indeed have an effect on the min-entropy level of an accelerometer's readings, this effect is dwarfed by the effect of physical movement on the sensor, even if this movement is as subtle as a stream of air from a fan, blow dryer, or other source.

4.6 Effect of Context and Users

All of the previous motion tests were conducted by directly handling WISP tags. In practice, however, many users do not directly manipulate their RFID tags. They instead leave their tags in their wallets, bag, purses, or other containers. These items are presented

to the RFID reader's antenna, allowing the tags to be read through the material of the container. Thus, we took an additional round of samples with the WISP tag placed inside different objects. First, the WISP tag was wrapped in bubble wrap and placed inside a cardboard box. Next, the tag was placed inside of a wallet. The wallet was tested both while placed open on a desk and held open while in front of the reader's antenna. The scenario where a tag is placed loose inside a purse or backpack was also tested.

The min-entropy measurements of these samples are provided in Table 5. In the case of the box, wallet, and backpack tests, the observed min-entropy estimates were actually 0.1 or 0.2 lower than the lowest min-entropy observed for the stationary samples that we recorded. This can be partially attributed to random differences between the two sample sets. However, the shielding tests conducted with vibration dampening pads discussed in Section 4.3 suggest that accelerometers are affected not only by small movements in adjacent solid objects but also by airborne vibration. We therefore conclude that these types of enclosures reduce the amount of detectable motion derived from both these sources by a small degree.

Movement	Min-Entropy
Box	3.3
Wallet	3.1
Hand Wallet	7.3
Purse	4.3
Backpack	3.3

Table 5: Min-entropy Estimates of Accelerometer Sample Distribution for 10 Minute Container Samples

Finally, several samples were also taken to test for variations between different users. All of the samples taken thus far were performed by the same test subject. While little variation was anticipated in the non-interactive samples, such as the stationary ones where a tag was left sitting on a desk, we wanted to make sure our tests captured any differences that might exist between the motions when performed by different volunteers. We therefore repeated the hand held and arc swipe tests with four different volunteers. These tests shed some light on the randomness of accelerometer readings under different circumstances. The min-entropy of these samples is given in Table 7 in the Appendix. The average value across all "volunteer hand" samples was 5.2 and the standard deviation of these measurements was 0.7. For the "volunteer swipe" samples, the average value was 8.8 while the standard deviation came to 0.3.

4.7 Effect of Resonance

We conducted a set of tests where an accelerometer was subjected to various types of tones in an attempt to cause resonance, as discussed in Section 3. To this end, we utilized a Creative Inspire 5.1 5300 speaker system [11] to output sounds of different frequencies. Since these speakers feature a 40 to 20,000 Hz operating range [10] they were well suited to subjecting acceleration sensors to different kinds of forces. We desired to use this audio equipment in conjunction with a WISP and its onboard ADXL330 accelerometer, but unfortunately the Impinj RFID reader interfered with the sound hardware when in use.

As an alternative, we utilized the LIS302DL accelerometer that is found on Nokia N97 mobile phones [38]. Recall that this model has a lower min-entropy than the ADXL330 accelerometer (refer to Section 5.2 for more details). Since this sensor has a resonance frequency of 2000 Hz [22], multiples of this frequency are the most likely to provoke feedback. We therefore attempted to create a resonance effect by playing tones with frequencies that were multiples of 500 Hz. Each tone was played at medium volume and a sample was taken for a duration of ten minutes.

The results of these tests showed no discernible correlation between the pitch of the tone being played and the min-entropy level exhibited by the device's accelerometer samples. These tests do not completely rule out the possibility of reducing the min-entropy of an accelerometer's output by inducing a resonance effect because it is certainly possible that we simply did not achieve the correct frequency. While we did produce the specified resonance frequency for the N97's accelerometer, perhaps this value was altered in practice by external elements such as the casing of the phone. Nonetheless, this result underscores the difficulty of creating such an effect even under ideal laboratory conditions.

5. DISCUSSION AND EXTENSIONS

5.1 Efficiency

While RFID read rates are notoriously difficult to measure in a reproducible fashion due to the number of variables involved, in the absence of a more standardized metric they will be used to gauge the plausibility of utilizing the approaches presented in this paper in a practical RFID deployment. The time between WISP reads over the course of our study was 31.2 milliseconds. 50 samples were needed to generate 128 random bits using the chained double extraction mechanism. It therefore takes $50 * 31.2$ milliseconds $= 1.6$ seconds to generate a single 128 bit random value using the chained double extraction mechanism. More generally, assuming an average accelerometer min-entropy contribution of 3.5 per sample, $k/3.5 + 20.2$ samples are required to produce a k bit output 2^{-35} close to uniform value. Combined with the observed sampling rate, this yields an execution time of $9.0k + 631.2$ milliseconds to generate a k random bits.

5.2 Mobile Phone RNG

We also took samples from an accelerometer on a mobile phone in order to demonstrate the applicability of this entropy collection technique to devices besides computational RFID tags. More specifically, we ran our tests on a Nokia N97 phone with a STMicroelectronics LIS302DL accelerometer. This is the same model that was utilized in our resonance experiments (see Section 4.7). We accessed this sensor using the J2ME Mobile Sensor API. We attempted to take overnight stationary samples using this device's accelerometer much as was done with our WISP tags, but for unknown reasons the phone consistently ceased logging after three hours. We therefore initially took two 3 hour LIS302DL samples.

The min-entropy of the first sample was estimated to be 1.1, while the second was 1.7. These estimates were significantly below those derived from our computational RFID tag samples. This is due to the reduced resolution of the LIS302DL in comparison with the ADXL330. The amount of min-entropy that accelerometers, and sensors in general, are capable of producing is a function of the device's resolution as we explain in Section 3. More sensitive devices are capable of picking up more minute variations in external phenomena and their readings will therefore capture more randomness. As a result, it makes sense that the N97's LIS302DL, with a resolution of 0.15328125 m/s^2, produces less entropy per reading than the WISP's ADXL330, which features a resolution of 0.05748046875 m/s^2. A complementary explanation of the reduced level of randomness experienced on the N97 is that its accelerometer is held steady by the other components surrounding it in the casing of the phone, while the WISP accelerometer component was left out in the open, exposing it to more variations in movement as a result.

As a final test of the N97's accelerometer, one of the authors performed a test where he carried the phone with him while performing his daily activities. The phone was set to log its accelerometer reading for the three hour limit while the tester treated it in precisely the same way as his actual cell phone. He kept the phone in his pocket while at his desk, eating a meal, and riding on mass transportation, lifted the phone to his ear when he received an incoming call on his actual phone, and held it under his real phone when sending text messages or surfing the web. The min-entropy of the phone's accelerometer readings did indeed increase dramatically when the tester used it to mimic daily usage. The estimate came to 6.3, a 4.5 time increase over the average of our stationary estimates. This proves that accelerometer based RNG is viable not only for highly constrained devices such as RFID tags, but also more general purpose wireless appliances such as cell phones.

6. COMPARISON: ACCELEROMETERS VS. OTHER ENTROPY SOURCES

We now argue that accelerometer based random generation is superior when weighed against prior state-of-the-art solutions. We accomplish this via a comparison with existing work on traditional and sensor based methods of entropy collection. See Figure 3 in the Appendix for a side-by-side comparison summary of the advantages and disadvantages of each entropy collection possibility.

6.1 Traditional Sources

In [15], Gutterman et al. establish that the Linux kernel collects entropy from four distinct sources: keyboard inputs, mouse gestures, hard drive use, and interrupt events.

Manual or Automatic? In order to register randomness, keyboards and mice must be moved in an unpredictable manner by a human user for the duration of the entropy creation process. Since humans are notoriously bad at behaving in a random fashion [18], this results in an unexpectedly high burden for users of RNG systems that utilize these interactive types of input. Hard drive events seem like a more promising RNG source than either mice or keyboards since they do not require explicit user involvement. Similarly, the use of radio events does not require any user interaction either. Interrupt events are vague and on many systems do not yield much entropy [15].

Found Where? While, as shown in Figure 3 of the Appendix, mice, keyboards, and hard drives are ubiquitous on desktop and laptop computers, they are uncommon on devices with a smaller form factor or more constraints such as RFID tags. Radio frequency noise is a natural choice as an entropy source for wireless devices since they are necessarily equipped with a radio receiver that could be used in this capacity.

Adversarial Control? Unfortunately, the susceptibility of wireless transmissions to outside manipulation makes them a poor choice for gathering entropy, as an adversary could easily overwhelm any existing radio noise by jamming the signal. A similar shortcoming of mice and keyboards is that the range of inputs that they register during normal operation is driven by the application in use at any given time. This means that they may contain much less entropy than expected or, even worse, potentially be predictable by an attacker. For example, when using a distributed application via a web browser, the information sent between the user's machine and the application server can provide detailed information about the locations of buttons and input fields that will be utilized. On web servers, we expect there to be a high volume of network traffic, and thus corresponding hard disk reads and writes, present. This is a good thing from the perspective of harvesting sufficient entropy. Unfortunately, much like mouse motion, the fact that this activity is driven by network traffic provides adversarial entities some level of control over disk activity.

Works When Stored? A device's mouse and keyboard cannot be used when the device is placed in a wallet or other type of storage, as shown in Figure 3 in the Appendix. Since mice and keyboards require constant user involvement to be able to craft entropy, they clearly cannot be used to this end while stowed. On the other hand, hard drives and radios do not require any user manipulation to function and are thus capable of achieving normal operation when placed in a bag, purse, or wallet.

Indefinite Reuse? Due to their limited use by a single individual the drives of standard desktop systems will be idle more often than not. Gutterman et al. found that an idle system generated only 16-bits of entropy every 15 minutes based on hard drive activity [15].

6.2 Microphones

Microphones are used to create randomness by the service provided at random.org [16].

Manual or Automatic? As Morrison [30] points out, microphones are preferable to mice due to the fact that mice require the devoted attention of a user while audio sensors do not. Like microphones, accelerometers are also sensors that require an analog to digital converter. However, microphones still require some user involvement because they must be set to a viable source of noise prior to use. Unlike microphones, accelerometers are ready for RNG without any user involvement whatsoever.

Found Where? Since microphones are a necessity for all mobile phones and are the most commonly encountered optional peripheral for desktop computers, they seem like a natural choice for use as a fount of entropy. This concept was further explored by Morrison in [30], where he points out that mice and microphones are the two common computer interfaces that utilize analog to digital converters. These are useful for RNG because the process of turning an analog signal into a digital value always introduces entropy into a system irrespective of the physical phenomenon that is actually captured by a sensor. A potential issue with sound based solutions is that they require the raw storage of sound files, which might take up too much storage space on constrained devices such as cell phones and RFID tags.

Adversarial Control? As shown in the Appendix's Figure 3, a random number generation technique that relies on a microphone is vulnerable to control, for example, by making loud noises. Morrison's work exposes this critical flaw with the use of microphones in the context of RNG. The output of his audio based randomness generator failed to pass statistical tests in cases where the sampling rate was too low as well as situations where the environment was either very quiet or noisy enough to cause the ADC to "clip," that is, exceed the range of the analog to digital converter.

Works When Stored? Microphones do not work when placed in a storage item due to the fact that any enclosure they are placed in will muffle ambient sound.

Indefinite Reuse? In general, as shown in Figure 3 which is found in the Appendix, microphones can be sampled repeatedly and indefinitely. The main problem Morrison found with using audio to derive entropy is that sound samples are correlated when sampled at a high rate, though. In order to avoid this, microphone samples can be added and sampled at a higher period. This decreases the correlation between consecutive samples, but unfortunately also reduces the output rate or the resultant random number generator.

6.3 Cameras

The next group of sensors that we turn our attention to are cameras. Bouda et al. elaborate on this intuitive choice [7].

Manual or Automatic? Like many of the other sensors listed in Figure 3 in the Appendix, cameras do not need manual intervention in order to take samples. If reliant on external data, however, an administrator must ensure that the camera in use is pointed at source that contains sufficient entropy, such as a lava lamp [31]. If, on the other hand, the camera based RNG technique does not require any external stimulus to operate, as is the case with the work of Bouda et al. [7], then no initial setup is required to instantiate a camera based random number generator.
Found Where? As listed in Figure 3 of the Appendix, cameras are found on a wide variety of devices.
Adversarial Control? If the external images captured by a camera were utilized as part of the entropy collection process, cameras would be vulnerable to manipulation. Bouda et al. sidestep this issue by relying solely on the mechanics of the camera for entropy rather than any external phenomenon. This is accomplished by sampling the camera while its shutter is closed. This scenario has limited applicability, however, as most web, laptop, and phone cameras do not have a shutter. The authors of this work show that one of the advantages of using a camera is that its samples yield a min-entropy of approximately 4.0. This is comparable to the accelerometer min-entropy estimates which are provided in Section 4. In addition, all of the sequences they tested pass 15/16 of the tests in the NIST battery, which is also comparable to our results.
Works When Stored? A camera can not collect external data when stored in a wallet or purse. Thus, if reliant on external data, a camera cannot be used for RNG when stored. If not, then it can be.
Indefinite Reuse? Similar to other sensors, cameras can be used indefinitely.

6.4 Other Sensors

In this subsection, we complete our analysis of alternative entropy sources by discussing the use of the remaining four sensors listed in Figure 3 in the Appendix. These are thermometers, photometers, proximity sensors, and magnetometers.
Manual or Automatic? As listed in the Appendix's Figure 3, all of these sensors are automatic with the exception of photometers which, like cameras, must be pointed at a light source with suitable variability to achieve RNG.
Found Where? Thermometers are frequently found on desktop and laptop machines as well as on some RFID tags, e.g., WISPs. Photometers are similar to cameras in that they are sensors of light, but unlike cameras, photometers are not found on any commercially available devices that we are aware of. Proximity sensors and magnetometers are starting to be deployed on cell phones and video game systems.
Adversarial Control? It is possible for adversarial control on all four of these sensor types to result in a loss of entropy. Thermometers can be exposed to a source of heat or cold that pushes their temperature beyond their operation range, for example. Along the same lines, photometers could be covered up and blocked from their randomness producing source of light. An item placed near a proximity sensor would cause it to constantly register the same value. Finally, a magnetometer could simply be moved to output a value of an adversary's choosing.
Works When Stored? The only miscellaneous sensor that works when stored is a magnetometer. Placing this device in an enclosure does not impact its ability to perceive magnetic fields.
Indefinite Reuse? The most beneficial part of these four entropy sources is that, as sensing hardware, they can be queried indefinitely for readings without any limitations.

6.5 Special Purpose Hardware

The generation of true randomness can be achieved by harvesting entropy from electrical and material processes within a device's own circuitry as opposed external phenomena [4]. This activity manifests itself in various forms, including thermal, shot, flicker, generation, and burst noise [4, 26, 27].

While hardware harbors internal unpredictability in the form of numerous varieties of noise, capturing this entropy and converting it into usable digital data is a non-trivial task. Devices require a mechanism through which they can sample minute and transient variations present in their own circuitry. Several different techniques for accomplishing this have been proposed, such as direct amplification and discrete-time chaos [4].

Random number generators that operate solely on internal entropy have some desirable characteristics. Since they do not need to perform any environmental sampling, their design is simpler than solutions involving sensors. This implies that their form factor can be smaller and their cost can be lower when compared to similar external techniques. Additionally, since they do not explicitly rely on sampling contextual phenomena, they have the potential to be more robust in the face of adversarial interference. On the other hand, since they do not involve any environmental monitoring, this class of techniques requires hardware that is necessarily single purpose in nature. As such, they may not be affordable for a given hardware design in terms of cost or space. Another downside to disregarding external entropy is that any randomness originating from beyond the device itself is forfeited, which may limit the amount of available entropy. Finally, throughput considerations may be an issue for users of internal random number generators. For example, while it is unlikely to suffer from protracted delays in practice, it is not possible to know whether or not the Intel random number generator will produce any output in a given time frame [4]. While useful when present, the specialized hardware needed to harvest internal entropy may not be available on any particular computing system. This is particularly true of low cost devices with small form factors, such as RFID tags. An accelerometer on an RFID tag, on the other hand, can be used for other tasks besides random number generation, such as context recognition as developed in [12].

7. CONCLUSIONS

In this paper, we established that an accelerometer is a source of entropy which possess some unique and appealing properties. Most importantly, we demonstrated that accelerometers, unlike other sensors, are resistant to a variety of environmental variations and even to adversarial manipulation. To support this claim, we developed a thorough experimental adversarial model for accelerometers when used as an entropy source. We also demonstrated that accelerometers compare positively to other entropy sources with respect to their universality and usability through a thorough comparative analysis. Furthermore, we showed that deriving entropy from an accelerometer should work on many devices by designing, implementing, and evaluating an accelerometer based RNG solution on the WISP computational RFID tag, which is a constrained device. Our experiments indicate that accelerometers generate sufficient entropy to meet some cryptographic needs even while stationary and produce even more when in motion. The best approach an attacker could take to interfering with the amount of min-entropy generated by an accelerometer would be to place one in as stable an environment as possible, as anything else will only serve to increase the min-entropy of the readings rather than reduce it.

Acknowledgments: We are thankful to our shepherd René Mayrhofer and WiSec'11 anonymous reviewers for their thoughtful feedback.

8. REFERENCES

[1] Mouser Electronics MMA7660FCR1 Freescale Semiconductor Board Mount Accelerometers. Available at http://www.mouser.com/search/ProductDetail.aspx?qs=uDmhV2jwPReFrqFV70kRUw==, 2009.

[2] Analog Devices. Adxl330 small, low power, 3-axis ±3 g imems accelerometer. Available at http://www.sparkfun.com/datasheets/Components/ADXL330_0.pdf, 2006.

[3] B. Crowell. Vibrations and Waves. Available at http://www.lightandmatter.com/html_books/3vw/ch02/ch02.html, 2009.

[4] B. Jun and P. Kocher. The Intel Random Number Generator. Available at http://www.cryptography.com/public/pdf/IntelRNG.pdf, 1999.

[5] B. Barak, R. Impagliazzo, and A. Wigderson. Extracting Randomness Using Few Independent Sources. In *SIAM Journal on Computing*, 2006.

[6] B. Barak, R. Shaltiel, and E. Tromer. True Random Number Generators Secure in a Changing Environment. In *Cryptographic Hardware and Embedded Systems*, 2003.

[7] J. Bouda, J. Krhovjak, V. Matyas, and P. Svenda. Towards True Random Number Generation in Mobile Environments. In *Nordic Conference on Secure IT Systems: Identity and Privacy in the Internet Age*, 2009.

[8] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall. Recognizing Daily Activities with RFID-Based Sensors. In *UbiComp*, 2009.

[9] E. Cochran, J. Lawrence, and C. Christensen. Quake-Catcher Network. Available at http://qcn.stanford.edu/, 2008.

[10] Creative Asia. Creative inspire m5300 5.1 speakers. Available at http://asia.creative.com/products/product.asp?category=4&subcategory=25&product=15999&nav=1&listby, 2010.

[11] Creative Worldwide Support. Technical specifications of creative 5.1 speakers. Available at http://support.creative.com/kb/ShowArticle.aspx?sid=47175, 2010.

[12] A. Czeskis, K. Koscher, J. Smith, and T. Kohno. RFIDs and Secret Handshakes: Defending Against Ghost-and-Leech Attacks and Unauthorized Reads with Context-Aware Communications. In *ACM Conference on Computer and Communications Security*, 2008.

[13] D. Yeager and A. Sample. WISP 4.1DL Schematic v8. Available at http://wisp.wikispaces.com/file/view/WISP4.1DL_Schematic_v8.pdf, 2010.

[14] H. Gilbert, M. Robshaw, and Y. Seurin. HB#: Increasing the Security and Efficiency of HB+. In *EuroCrypt*, 2008.

[15] Z. Gutterman, B. Pinkas, and T. Reinman. Analysis of the Linux Random Number Generator. In *Symposium on Security and Privacy*, 2006.

[16] M. Haahr. RANDOM.ORG - True Random Number Service. Available at http://www.random.org/, 2010.

[17] J. Halderman, S. Schoen, N. Heninger, W. Clarkson, W. Paul, J. Calandrino, A. Feldman, J. Apelbaum, and E. Felten. Least We Remember: Cold Boot Attacks on Encryption Keys. In *USENIX Security Symposium*, 2008.

[18] R. Halprin and M. Naor. Games for Extracting Randomness. In *Symposium On Usable Privacy and Security*, 2009.

[19] D. Holcomb, W. Burleson, and K. Fu. Initial SRAM State as a Fingerprint and Source of True Random Numbers for RFID Tags. In *Conference on RFID Security*, 2007.

[20] D. E. Holcomb, W. P. Burleson, and K. Fu. Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *IEEE Transactions on Computers*, 2009. to appear.

[21] J. Holleman, D. Yeager, R. Prasad, J. Smith, and B. Otis. NeuralWISP: An Energy-Harvesting Wireless Neural Interface with 1-m Range. In *BioCAS*, 2008.

[22] ICBuy.com. LIS302DL Acceleorometer Specifications. Available at http://tec.icbuy.com/product/productView/id/162826.html, 2008.

[23] B. Jiang, S. Roy, K. Sundara-Rajan, M. Philipose, J. Smith, and A. Mamishev. Energy Scavenging for Inductively Coupled Passive RFID Systems. In *IEEE Instrumentation and Measurement Technology Conference*, 2005.

[24] B. Jiang, J. Smith, M. Philipose, S. Roy, K. Sundara-Rajan, and A. Mamishev. Energy scavenging for inductively coupled passive RFID systems. In *IEEE Transactions on Instrumentation and Measurement*, 2007.

[25] A. Juels and S. Weis. Authenticating Pervasive Devices with Human Protocols. In *CRYPTO*, 2005.

[26] K. Lundberg. Noise Sources in Bulk CMOS. Available at http://web.mit.edu/klund/www/papers/UNP_noise.pdf, 2002.

[27] M. Tormanen. Analog IC Design 2010: Lecture 9 - Noise. Available at http://framtiden.eit.lth.se/fileadmin/eit/courses/eti063/lectures2010/AnalogIC_F9.pdf, 2010.

[28] MEMS Industry Group. Nokia Beats Apple to Compass-in-Phone. Available at http://memsblog.wordpress.com/2009/12/03/nokia-beats-apple-to-compass-in-phone/, 2009.

[29] D. Molnar and D. Wagner. Privacy and Security in Library RFID: Issues, Practices, and Architectures. In *ACM Computer and Communications Security*, 2004.

[30] R. Morrison. Design of a True Random Number Generator Using Audio Input. In *Journal of Craptology*, 2001.

[31] L. Noll, S. Cooper, and M. Pleasant. LavaRnd. Available at http://www.lavarnd.org, 2003.

[32] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. In *Special Publication 800-22*, Available at csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html, 2008.

[33] A. Sample, D. Yeager, P. Powledge, and J. Smith. Design of a Passively-Powered, Programmable Sensing Platform for UHF RFID Systems. In *IEEE International Conference on RFID*, 2007.

[34] N. Saxena and J. Voris. We Can Remember It for You Wholesale: Implications of Data Remanence on the Use of RAM for True Random Number Generation on RFID Tags. In *Conference on RFID Security*, 2009.

[35] N. Segawa. Behavior Evaluation of Sika Deer (Cervus Nippon) by RFID System. In *WISP Summit*, 2009.

[36] S. Skorobogatov. Low Temperature Data Remanence in Static RAM. Available at www.cl.cam.ac.uk/techreports/UCAM-CL-TR-536.html, 2002.

[37] J. Smith, A. Sample, P. Powledge, A. Mamishev, and S. Roy. A Wirelessly-Powered Platform for Sensing and Computation. In *8th International Conference on Ubiquitous Computing*, 2006.

[38] STMicroelectronics. LIS302DL MEMS motion sensor 3-axis - 2g/8g smart digital output "piccolo" accelerometer. Available at http://www.st.com/stonline/products/literature/ds/12726/lis302dl.pdf, 2008.

APPENDIX: Additional Tables and Figures

Figure 2: Effect of Altering the WISP Read Rate on the Min-Entropy of its Accelerometer Samples

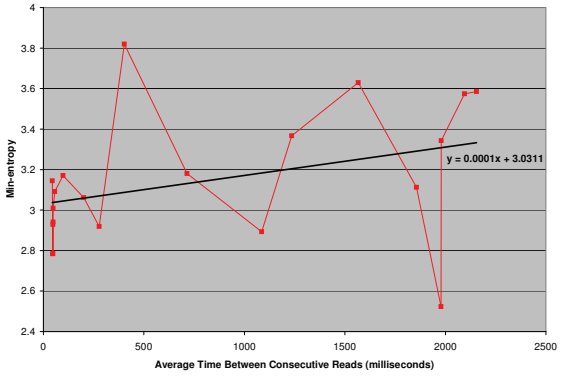

Figure 3: Comparison Table (highlighted cells represent positive features)

Sensor or traditional?	Entropy source	Manual or automatic?	Found where?	Adversarial control?	Works when stored?	Indefinite reuse?
Sensor	Accelerometer	Automatic	Cell phones, certain laptop models, certain video game remote control models, fitness aids, WISPs	Can only increase entropy	Yes	Yes
Traditional	Mouse	Manual	All laptops, all desktops, certain cell phone models, and certain gaming system models	Can decrease entropy	No	No
	Keyboard	Manual	All laptops, all desktops, certain cell phone models, certain gaming system models	Can decrease entropy	No	No
Traditional	Hard Drive	Automatic	All desktops, most laptop models, most gaming system models	Can decrease entropy	Yes	No
	Radio	Automatic	Cell phones, laptops, video game remote controls, portable gaming systems, optional desktop peripheral, routers, WISPs	Can decrease entropy	Yes	No
Sensor	Microphone	Requires initial setup	Cell phones, portable gaming systems, optional desktop peripheral	Can decrease entropy	No	Yes
	Camera	Requires initial setup (if reliant on external data)	Cell phones, certain laptop models, certain monitor models, certain gaming system models, optional desktop peripheral	Can decrease entropy (if reliant on external data)	No if reliant on external data, yes otherwise	Yes
	Thermometer	Automatic	Desktops, laptops, WISPs	Can decrease entropy	No	Yes
	Photometer	Requires initial setup	Uncommon on commercial devices	Can decrease entropy	No	Yes
	Proximity	Automatic	Certain cell phone models, certain gaming system models	Can decrease entropy	No	Yes
	Magnetometer	Automatic	Certain cell phone models	Can decrease entropy	Yes	Yes

Table 6: Temperatures (in degrees Celsius) and Min-Entropy Estimates for Temperature Control Samples

Control Method	Min. Temp.	Average Temp.	Max. Temp.	Min-Entropy
Blow Dryer	40.5	69.8	81.4	6.6
Freezer	14.9	26.3	27.7	3.8
Fan	21.1	22.4	22.7	6.0
Heater	28.1	32.3	35.1	4.8
Thermos	-1.2	-0.9	0.0	4.7

Table 7: Min-entropy Estimates of Accelerometer Sample Distribution for Multiple Volunteer Tests

Movement	Min-Entropy
Volunteer Hand #1	5.5
Volunteer Hand #2	3.9
Volunteer Hand #3	5.4
Volunteer Hand #4	5.8
Volunteer Swipe #1	8.5
Volunteer Swipe #2	9.1
Volunteer Swipe #3	8.7
Volunteer Swipe #4	9.1
Sitting Still	4.7
Sitting Shaking	8.5
Walking	10.9
Jogging	11.1