5-2011

Protocol Design for Sensor Networks

Harry Tian Gao
College of William and Mary

Protocol Design for Sensor Networks

A thesis submitted in partial fulfillment of the requirements for a
Bachelor of Science with Honors in Computer Science
from the College of William & Mary in Virginia,

by

Harry Tian Gao

Accepted for: _____

_____
Qun Li, Director

_____
Michael Lewis

_____
Weizhen Mao

_____
Virginia Torczon

Williamsburg, Virginia
April 27, 2011

Abstract

This thesis presents the difficulties and solutions to local processing, secure transmission, and central processing, which are the three stages of sensor network protocols. The thesis is organized into three chapters, each focusing on a particular real-life application that highlights the challenges of a particular stage. The first chapter uses body sensor network as an example to show the importance of local processing, the second chapter deals with vehicular network to illustrate the complexity of secure transmission of information, and the third chapter uses location proof service via RFID tags to demonstrate the difficulties and possibilities of central processing and data consolidation.

Acknowledgments

I would like to take this chance to thank Professor Qun Li for his continuous guidance throughout the my college career and his active involvement in all the research projects associated with this thesis. He motivated me to tackle one research topic after another, creating an uniquely fulfilling experience.

I would also like to thank Professor Robert Lewis for his support and help on the projects. His analysis provided the mathematical rigor for the RFID project to excel. In addition, the projects put forth here are not just the fruit of my labor, but the collective intelligence and effort of many individuals, especially Seth Utecht, Fengyuan Xu, Haodong Wang, and Andrew Wilcox.

Many thanks to Professors Weizhen Mao and Virginia Torczon for agreeing to be on my honors committee, the CSUMS faculty and staff for a wonderful experience which enhanced my understanding and interest in sensor networks, and US National Science Foundation grants CNS-0721443, CNS-0831904, CAREER Award CNS-0747108, and CSUMS grant DMS 0703532 for their funding.

Finally, I am grateful to Jessica Miller for proofreading and moral support.

Contents

Part I

Introduction

This thesis focuses on wireless sensor networks' protocol designs. Sensors are small, inexpensive devices that can unobtrusively provide solutions to a wide range of problems, many of which are already commonplace. Sensors play an important role in many fields, such as medical, home security, and motion detection. Their future is also promising as more and more science fiction theories turn into real life applications.

Before looking into the details of their operations, we must understand the hardware we are working with – their capacities and their limitations. Typically the word "sensor" refers to sensor nodes known as motes (Fig.1) that operate on batteries and

Figure 1: The particular mote used to carry out the experiment on the body sensor network [1].

have some processing power, and can gather information from the environment with embedded sensors (e.g. accelerometer, potentiometer etc.). They can also communicate with one another through radio signals. The non-volatile memory on these devices is limited but nontrivial. One of the most significant considerations for these sensors is battery usage.

Another type of sensor, known as passive radio frequency identification (RFID) tags (Fig.2), face a different set of challenges. The ones we are interested in are passive tags; that is to say, unlike active tags or motes, they do not rely on batteries and are unable to measure, compute, or otherwise engage with any of their surroundings. They can only respond when a reader initiates communications, in which case the reader's external electromagnetic field will temporarily power the tags. Such simple architecture makes RFID tags inexpensive, with very low computational power and

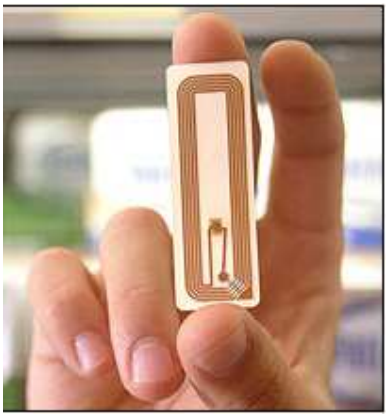


Figure 2: Passive RFID tags are tiny and inexpensive.

virtually no memory. Because the device is completely "off" except when readers are nearby, it is impossible to keep a running clock, which can be problematic as demonstrated in the RFID chapter of this thesis.

We identified three major stages to a successful streamlined usage of sensors to solve any given problem:

1. Measure data, process data locally, and extract the useful information;
2. Securely transfer the data to a central processing unit; and
3. Consolidate and interpret all the data to provide a useful solution.

At each stage, there are key considerations and challenges. Local processing and data selection allow the system to extract crucial information at an early stage, reducing the burden on the delivery and consolidation mechanisms. However, it is not obvious what information is essential and what can be safely ignored, especially without knowing what other local sensors are collecting.

For the delivery stage, security is key. Since all important data should be routed to the central processor via some delivery medium, a compromised transportation mechanism can result in a seriously faulty system, where a malicious user can intercept and forge information at will.

Lastly, the central processing unit must consolidate all the information fetched from local sensors, and make sense of the individual pieces of the puzzle. Depending on the application, this process can be extremely complicated since there is no guarantee that the sensor data will be correct and consistent with one another.

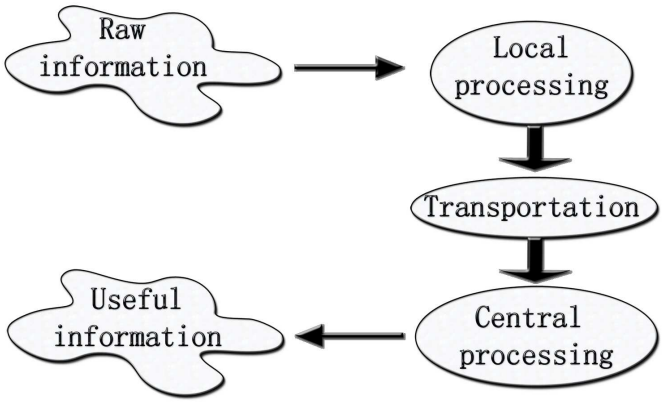


Figure 3: Sensor network protocols gather from a large set of raw information to produce a small but useful set of organized information.

Together, these three stages can convert a large amount of raw information into useful information (Fig. 3). In order to explore the challenges involved at each stage outlined above, this thesis presents solutions to three real life applications, highlighting the intricacies of each of the three stages. The first two applications use motes as the basic hardware, while the third uses RFID tags. All three applications rely on sensors to collect data to help infer conditions about the environment and/or users' states. The array of solutions utilizes both types of hardware introduced earlier, namely motes and smart tags.

Body Sensor Network The ability to identify basic human motions has a wide range of applications. From activity monitoring to fall detection, accuracy and efficiency are both crucial to a successful design. In the body sensor network chapter, we present an energy-conscious protocol that still enables us to correctly identify various motions. With an average experimental result of 97% accuracy, we hope the design will provide a simple and elegant solution to basic motion detection needs. We attached motes to the human body to measure the physical activities of different parts of the body. The information was then transmitted to the central location, e.g. a home computer, and the user's motion could be inferred from the movement of different parts of the body. Without an elegant solution, sensors must send out a large amount of data to the central server, but doing so would quickly drain the sensors' batteries and make the protocol impractical. Therefore, efficient local processing and selective uploading

are instrumental to a successful protocol.

Vehicular Network The next chapter focuses on efficient and secure delivery of information collected and processed by the local sensors to the central server. This poses significant challenges in the field of vehicular sensor network. We consider a simple vehicular network infrastructure composed of roadside sensors and vehicles. We show, through a simple secure symmetric key based protocol design and its implementation, the feasibility of secure data collection in vehicular sensor networks. Sensors are deployed to monitor the road environment, and the accumulated data can be collected by the vehicles passing by.

On the one hand, there is a small window during which the sensor-in-transit can communicate with the roadside sensors, which forces us to make the secure handshake as simple as possible to allow sufficient time to harvest the data. On the other hand, the security protocol must be sophisticated enough to defend against common tactics the malicious party may deploy to compromise the data. We conducted experimental studies to demonstrate that the protocol works in a realistic setting by collecting the real trace data through real implementation. Some of the key considerations are efficiency, usability, and security. The protocol does not safeguard against some of the techniques an adversary could deploy, such as jamming and deliberate battery-draining.

Location Proof In the next chapter, we use RFID tags rather than motes as the basic hardware to maximize the difficulty in consolidating the pool of data from local sensors. The simplicity of RFID tags makes it impossible to perform extensive local processing, which increases the burden of the central server. In particular, we attempt to provide location proof service to clients by deploying RFID tags. The premises of location proof service are very intuitive. Whether it is a mall's reward program where customers wish to show they have spent a large amount of time in the stores, or mission critical information only available to people who are physically present at a given location, location proof is useful in a variety of situations. Current methods tend to be expensive, labor intensive or intrusive. For example, closed-circuit television (CCTV) has been a traditional method for many years, but the cost of equipment, setup and monitoring make it increasingly unattractive in today's fast paced world.

The location proof solution provided in the last chapter relies on self-reporting,

and it opens up vulnerability for the malicious party to inject directly falsified data into the system. It is then up to the central server to consolidate a large amount of information and intelligently derive the correct information. This situation relates to the worst-case scenario, where either the data collection mechanism or the data transportation mechanism is compromised, and it is interesting to see what is possible for the central server to do, and how much knowledge it can extract in this situation.

Together, this thesis presents five chapters, three of which focus on different applications in which sensor networks play a crucial role in providing secure and efficient solutions. Those applications outline the diverse yet unified field of sensor networks, and help reveal the challenges and solutions associated with the protocol design process. The three applications also provide us a better understanding of the process by amplifying the challenges in each of the three stages, and the applications are useful in their own right as they solve interesting real life problems.

Part II

Body Sensor Network

This chapter focuses on using motes to provide motion detection for body sensor networks. It highlights the difficulties associated with local processing, especially in terms of efficiency and power management. We hope that it provides insight into not only this one application, but local processing protocols in general.

In recent years, developments in the field of body sensor network have shown many promises that can realize a wide range of goals. The measured and recorded data could be significant for personal, medical, entertainment, and legal purposes. Some applications, such as fall monitoring for the elderly, are already commercially available. More advanced topics, such as motion detection based on a network of sensors, are seen as more interesting and promising by the academic community [2, 3].

While the technology is not yet fully mature, engineering efforts to miniaturize sensors, the need for more accurate motion identification, and the incorporation of electronic devices into people's daily lives are just a few factors that suggest a growing possibility. One key issue haunting many of the implementations is the need for an energy-efficient design, because frequent battery change would be a hassle for the potential users, hence greatly hindering the usefulness of protocols.

In this study, we present a very energy-efficient design of a body sensor network, its implementation, and its evaluation. One central goal is to minimize the amount of data the sensors need to transmit to the base station. Since radio communication contributes to the greater part of the power consumed — the energy cost of transmitting 1KB of data over a distance of 100m is about the same as executing 3 million instructions on a MIPS processor [4], it is necessary for any energy-conscious design to limit the transmission rate.

This need for less power consumption weighs against the quality of service (QoS), because it requires intensive and frequent transmission of data in order to provide real-time analysis and to capture detailed data, while gathering and transmitting data at an infrequent rate would lower the resolution and accuracy of the design. For example, the energy saving scheme deployed by Ganti et al. in SATIRE [5] puts the sensors into a low-power mode after a period of inactivity, during which the sensors periodically check for activities and wake up to operate in normal mode when user

activity is detected. However, if a sleeping elderly wearing the monitor (which would be inactive) suddenly rolls off the bed, it is likely that the sensors will be sleeping and hence fail to pick up the signal. In fact, if the elderly remains motionless after the fall, the sensors will never know that he had fallen. This possibility will lower the QoS for such energy-saving designs.

This chapter attempts to present a design with both accuracy and efficiency in mind, and strives to create a system where the user's motions can be correctly identified by examining a set of accelerometer readings in real-time, as well as to record and reconstruct the user's activities. Moreover, because it is a hassle to train the system prior to use, the design will ideally work out-of-the-box, which is equivalent to designing a system where motions can be accurately detected using only the training data from different individuals. The system can also be used to detect abnormal motions, providing an early warning for possible need of physical therapy. The design can satisfy many possibilities envisioned in the field of body sensor network, but will not attempt to provide identity protection or serve to provide protection against malicious users who wish to confuse the system and disrupt the normal data collection procedures. However, it does offer a simple add-on protocol that can address basic security needs.

In this design, the motes process the accelerometer data locally and send acceleration data to the base station in terms of the gravitational constant g, or about 9.8 m/s^2. All acceleration greater than 7g is represented as 7g, because it is unlikely for a normal individual to sustain acceleration greater than 7g in any kind of activity. This allows the 2-axis acceleration data from the 5 different motes to be represented by two 3-bit numbers each, which reduces the package size significantly. Two other energy-saving strategies, local max delivery (LMD) and assumed continuity, are implemented in the systems as well and are discussed in detail in the efficiency section.

The data transmitted to the base station is analyzed and classified either as one of the pre-defined motions or *undefined*. Five pairs of data from the 2-axis accelerometers form a 10-number state array, which is used to identify the current motion. Assuming the base station is in range, the protocol will provide real-time motion detection and monitoring. If such capacity is unnecessary or infeasible due to specific environmental constraints, the motes can store data locally to transmit later. If the motes are aware that transmission is impossible without attempting to communicate by inferring from the received signal strength indication (RSSI) of previous packets, power will be

further conserved. The accuracy of the design is tested through a series of experiments presented in the experimental result section.

1 Related Work

Many possibilities and potentials have already been envisioned and/or implemented. From using motion capture technology as an input interface [6], to detailed monitoring of users' actions [7], a wide range of hardware has been used to provide innovative ways to better meet our needs. Many of the models focus on either specific body parts, especially arms and hands [6, 8, 9], or focus on a specific environment, such as driving a car [10], rather than attempting a generic, everyday motion identification and capturing scheme.

While some researchers try to take advantage of the fine differences between individuals [11], which allows for gesture-based personal identification, those differences impose a challenge on others. For example, in SATIRE, it was found that without user-specific training, activities like reading and typing become impossible to identify accurately [5].

The hidden Markov model (HMM) is a popular choice for previous work done in the field [6, 5, 9], but it has several shortcomings. For instance, it has a lower accuracy when trying to determine the current motion in real-time, and its performance worsens exponentially as the number of possible motions increase. In fact, because of the heavy reliance on the correctness of previous identifications, the HMM could potentially carry forward mistakes for a prolonged period after one misclassification. Therefore, HMM is best suit for the reconstruction of motions, not real-time monitoring. As such, HMM may not be able to satisfy some of the live monitoring needs.

Researchers in the medical and physical therapy fields have seen high potential in the use of body sensor network, allowing for remote emergency and administrative care [12]. Some wearable sensor networks utilize a number of sensors to collect different types of data [13] and attempt to determine the physical well being of the user by factoring all the relevant information. However, like other similar designs, there is a high battery usage and it may require frequent battery replacement or recharging, which would be very inconvenient for the end-user. This becomes especially problematic for products targeting elderly, who may already have difficulties carrying out other daily activities.

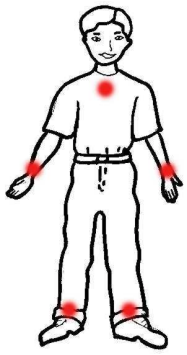


Figure 4: In this figure, the placement of the five motes attached to the body are marked with dots.

2 Problem Formulation

The design and associated protocol this chapter describes attempts to fulfill the needs outlined in the following scenario.

2.1 Problem Setting

A user wishes to identify, monitor, and/or record his daily actions via motion capturing technology. The system should include detection sensors that are most likely embedded into a suit or several pieces of clothing; another sensor that functions as the base station; and a processing device, such as a computer or cell phone, where the processing device is connected to the base station and meets the computational and storage needs. The user would like to have real-time monitoring wherever possible. The real time monitoring may be essential for retirees, firefighters, police, and many others. The sensors will detect, process and transmit data on horizontal and vertical acceleration of different parts of a human body. The user wishes to have the ability to use the system out-of-the-box, without the need for complicated and individual training, although it should still be possible for users to customize if they so choose.

2.2 System Design

- Five sensors will be used to measure the acceleration of the wrists, ankles, and chest (see Fig. 4).
- The sensors will measure and record the acceleration at a frequency of 10Hz,

and send out data at a frequency up to 2Hz. Each data pocket will contain two data points, one measuring the acceleration in the X direction, one in the Y direction. A processing device will be connected to the base mote, which receives the communications from the body sensors attached to the user and then relays it to the processing device. The device will use the model to identify the current motion of the user in real-time. Because the computational intensity required to identify the motion is not high, the device does not have to be a computer. In fact, a smart phone might work better in some cases, because it is more likely for the user to be in the vicinity of his cell phone than of his computer. However, cell phones usually have limited storage capacity; therefore, when a cell phone is used as the processing device, it may need to periodically dump all the data onto a computer or server via the cellular network.

- The motion identification system is expected to function out-of-the-box using one of the pre-defined settings. For example, the system may ship with a CD-ROM, on which there are many sets of models trained by different groups of people. Users can pick the gender, age group, and physical fitness most akin to their own conditions to insure the best performance. Because the models only affect the processing device and not any of the motes, there would be no need to plug each mote in and update its firmware. If a setting does not seem to be able to accurately determine the user's actions, or if a user's stats are dramatically changed (maybe because the user twisted an ankle and can not use one foot for a while), the user can always change the settings and continue to use the system without much hassle. This way, the end-user would not have to tediously train the system before usage, and does not need to retrain the system when his ankle twists. However, if an user prefers the system would also allow for customization and personalized training for better accuracy, he should be able to do so.

- The system should be highly efficient in order to avoid frequent battery changes. It should also be able to cope with the possibility that a user may be out of range with respect to the base processing device. While clearly under such circumstances it is impossible to identify the user's actions in real-time, the processing device should be able to reconstruct the motions of the user when the user comes back into communication range.

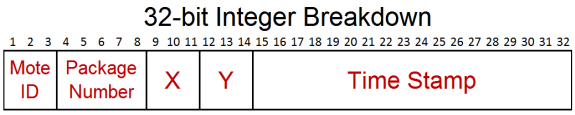


Figure 5: The breakdown of the single integer containing all the information a sensor collects. X and Y correspond to the data collected from the motes' two-directional accelerometer.

2.3 Goals

This chapter further attempts to determine the current motion of someone wearing the motes in real-time, not just the reconstruction thereof, while possessing the following characteristics:

- **Efficient**. The most important characteristic for a realistic design is efficiency, because frequent battery changes would be an annoyance to the end-user, which in turn dramatically lowers the QoS, as well as increasing the maintenance cost for the system. Apart from some standard and low-level ways to conserve energy, the proposed system deploys three key strategies that aim to further reduce the operational energy cost:

  - *Single Integer Messages (SIM)*. Instead of sending the raw data of analog voltages back to the base mote, a quick local processing is done to calculate the magnitudes of X- and Y- direction acceleration in terms of the gravitational constant, g. Experimental results show the acceleration is within 0 and 7 in virtually all cases, which can then be represented by only 3 bits of data. It is sent to the base along with mote ID (3 bit), package number (5 bit) and a time stamp (18 bit). Together, these pieces of information form one single 32-bit integer (See Fig. 5).

    The time stamp has a resolution of 0.5 second, same as the maximum frequency of data transmission. With 18 bits, there will not be any ambiguity so long as two consecutive packets are less than 36.4 hours apart. Given that a packet is created every time a change in acceleration occurs (see Assumed Continuity below for details), it is reasonable to assume that the gap is much smaller.

  - *Local Maximum Delivery (LMD)*. Operating at 10 Hz, the mote's accelerometer collects 5 data points within a half second time frame. Instead

Figure 6: In this figure, the leg in the front has 0 acceleration for a short time; the leg in the back, however, is accelerating forward until it passes the front leg, then it will decelerate and stop moving altogether as the other leg picks up speed again.

of transmitting all 5 data points, only the maximum of the 5 is sent; this is because the experiments show that only the greatest acceleration within a short time interval is interesting for the purpose of motion identification. By eliminating the less useful data, we managed to send data at a rate no higher than 2Hz without losing crucial data. Logically, it is not hard to understand why only the highest acceleration matters within a short time frame. Consider a very active motion, such as *running*. The acceleration on any particular part of the body, such as the leg, is not constantly high. It is actually an irregular *wave* function that alternates between an absolute value of 0 and some positive value (see Fig. 6). Only the highest positive value obtained is useful for analyzing and determining the motion, because that is the data differentiating *running* from, say, *walking*. In fact, because of the similarity between the two motions, the only real difference is the peak acceleration achieved.

– *Assumed Continuity.* After calculating the local maximum for the current 0.5 second time interval, the body motes compare the new x- and y-acceleration data with the data from the last 0.5 second time frame. If they are equal, that is, if there is no change in the acceleration in either direction, no new data is sent to the base. The base, in turn, assumes that the previous state of the mote persists if no new data is received. This allows for a further reduction in the frequency of radio communications. For example, if the user is in an elevator which has a constant acceleration for 3 seconds, then the motes will only send out the accelerations they measure once, for the first half-second time interval, rather than transmit-

ting 6 packets with similar contents at a rate of 2Hz. This is especially useful because users typically remain still for extended periods of time [5]. When someone is sleeping, for instance, the motes may only transmit dozens or at most hundreds of packets recording the scarce user movement during sleep, as opposed to sending out uninteresting data constantly at a rate of 2Hz. This reduces the number of packets by tens of thousands a night. However, unlike other energy saving strategies, the motes remain fully functional during this time and do not operate in a lower-frequency mode in order to save power; Therefore, the motes remain fully capable of monitoring sudden significant changes, such as an elderly rolling off the bed.

- **Accurate**. A design is clearly only useful when it can accurately determine the motion a person is performing. We hope to achieve an overall accuracy of at least 90%, and an overall identification rate above 90% as well. We believe that it is better to not make identification than to make the wrong one; consequently, the system is less aggressive and is more willing to make no identification when the state the user is in is ambiguous rather than giving its best guess. However, this is purely an engineering consideration and this parameter, like many others used in the design, can be tweaked to better meet the needs of the specific application.

- **Expandable**. While only 6 motions are defined in the current implementation of the protocol, we kept the code generic enough for easy expansion. We envision a complicated system with hundreds or even thousands of motions defined and we expect the additions to require minimum changes to the system without any conflicts within the current implementation. The only change required are to train the model with new motions, and such a change would be done only on the processing unit's side, with no modification on the motes. Therefore, it is possible for the system to be deployed with a set of standard motions, but allow users to add/drop any additional ones (the data of which will either be on the CD shipped with the product or be DLC) as they see fit. This flexibility provides additional functionality and allows for specialization and customization unmatched by other similar systems that have been proposed[5].

- **Combinable**. While the system can operate alone, its advantages can be fully

exploited when combined with other hardware that may provide more information. To make the combination seamless, the protocol measures the bodily motion of the monitored person without imposing any limitations. We envision the system to work alongside other technologies. For instance, while the design may be able to tell if someone is currently reading, it is impossible to tell what book he/she is reading, where he is reading, and who else is with him. If such information is important to the user, then the protocol needs to be incorporated into a context-aware tagged environment. Separately, the processing unit can realize that the user is reading from the system outlined in this chapter, and can know that a particular book is very close to the user by reading the RFID on the book [14].

- **Universal**. Much of the previous works done in the field require user-specific training. As seen in [5], many activities are unrecognizable for a user when the system is trained with another user's data. This means that an end-user would have to train the system him/herself before using the system. This could become a serious problem when a large number of motions are defined. Even if each motion only require minutes of training, it would still take hours of exhausting training, which many users may be unwilling to do. Worse yet, some motions, such as falling, cannot be easily trained. It is unlikely and unreasonable to ask any person to trip and fall down dozens of times just so that, next time the user falls, the system can identify it.

  Therefore, one of the goals of the design is to avoid user-specific systems, so one person's data would be applicable to another person's so long as they have roughly similar physical characteristics. This allows the monitoring system to be delivered to the end user with a set of preset settings (child, young adult, middle-aged adult, elderly adult etc.), and the system would then be ready to use after the user simply selects a group that best describes his or her characteristics.

3 Protocol

The following protocol is designed in order to achieve the aforementioned goals.

3.1 Local Processing

The data collected by x-axis and y-axis accelerometers at 10 Hz are locally processed before transmitting to the base to minimize the amount of radio communication while retaining crucial information. The following algorithm is used:

1. Collect 5 data points; keep the maximum value and discard the rest.

2. Convert ADC units to g for the data point. For the particular motes we used, the documentation states that

$$a = \lceil (v - 2048)/245 \rceil \tag{1}$$

   where v is the ADC measurement and a is the number of g's.

3. If both x-axis and y-axis values are the same as the previous values sent, disregard the data pair; otherwise, proceed to prepare for transmission by creating the SIM packet.

   - Note that a time stamp is only necessary if the motes' wearer might get out of the range of the base. If, for example, when the base is attached to a cell phone, which the wearer carries with him at all times, the time stamp would not be necessary, and the size of the packet can be even smaller–14 bits will do the trick.

4. Remove the saved SIM with the same ID number; store the new SIM locally in its place. Go to step (1).

3.2 Central Processing

The base mote will forward the SIM to the processing device, which will then determine the motion using a best-fit model with particular weighted penalty scheme and ideal values obtained from the training data. When one of the motes' data is received, the old data set is updated; data from the other motes remain untouched and are assumed to be the same as before unless/until new data come in.

Model

- *Training*

To train the model, there must be a pool of data for each predefined motion. The data will consist a set of arrays, each array made up the acceleration data from the x- and y-axis acceleration of the body mote. By examining the set of array as 10 distinct sets of points, we can find the mean of each set and compose the 10-points array of ideal values for each particular motion. Similarly, we can find the standard deviations for every motion predefined. The standard deviation provides an interpretation of the amount of penalization. For instance, if the training data for *running* shows that the chest x-acceleration is 1.2 with a standard deviation of 0.4, and a particular motion has a chest x-acceleration of 2, then it is two standard deviations away and hence should suffer a greater penalty than if the standard deviation was 0.8. As a result, motions with a more dynamic range of acceleration data will be less disturbed by absolute mismatch, while motions with small standard deviations will be much more susceptible to noise in the data. An extreme case of this is the motion *still*, which has a zero-tolerance – the standard deviation for all ten mote readings are zero, so a non-zero value would be enough to cause a motion to be defined as something other than *still*.

- *Basic Operations* At any given time, the model has to classify an array of 10 up-to-date data points describing the current motion. It will do so by comparing the motion to the pool of predefined motions and look for the one with ideal values that most closely approximate the current readings. In particular, the model will compute the following penalization function for all predefined motions

$$p = \sum_{i=1}^{10} \frac{(V_i - I_i)^2}{D_i + a},\ a > 0 \tag{2}$$

  where p is the total penalty for a particular motion, V is the 10-number array of the current unknown motion, I is the ideal values for a particular motion, D is the deviation values for the same motion, and a is a positive parameter value. Adjusting a can help minimize the error in prediction, and it helps avoid division by zero when the standard deviation is 0. The motion, which produces the lowest penalty value, is the one that most approximates the current motion, so the current motion will be identified as either that motion or as unknown.

- *Unknown Motions* It is possible for the current motion to be very dissimilar

to all the predefined motions. In that case, instead of identifying the current motion as the most similar predefined motion, the model simply recognizes the motion as unknown. In other words, if all p in Eq. (2) are greater than the parameter P_{max}, then the model does not identify the current motion as any of the predefined ones. The maximum threshold can be adjusted up or down to make the model more aggressive or conservative, respectively.

- *Continuity Bias* Fluctuation in the user's acceleration data within a half second time frame is smoothed out with LMD, but fluctuations over a longer time frame still hinders the accuracy of the model. At times, the model can make the wrong identification momentarily before going back to the correct one. If, for example, a walker speeds up to cross the street for a second or two, the penalty value for *walking* may increase slightly as the penalty for *jogging* decreases, and the latter may temporarily becomes the smaller of the two values. As a result, the model still identifies *jogging* briefly before going back to *walking*. Such brief change is often uninteresting to the user, and it may be better to represent the entire walk as *walking* rather than having brief episodes of other interrupting motions. However, the model cannot simply ignore the brief abnormality in the continuation of a motion. If, for example, someone is sleeping and abruptly falls off the bed, it would be quite erroneous to ignore the brief interruption of the *still*. Instead, the model provides some continuity bias to the current motion, making it more likely to identify the current motion as the same as before, but only enough to maintain the current motion in the face of small quantitative changes, not abrupt qualitative ones. If the current motion has a penalty of p_c, then the model will identify the current motion as the one with the adjusted penalty of

$$min(p_c - \tau, P_m, p_j, |j = 1, 2..n) \tag{3}$$

where τ is the continuity bias, P_m is the maximum penalty allowed for a motion identification, and p_j the penalty of the jth predefined motion. n is the total number of motions predefined.

Out of Range

When the signal strength, as indicated by the RSSI, falls below a certain threshold, the base signals all motes that the out-of-range mechanism is activated and all SIMs

are stored locally. Normal operation resumes when the motes once again come into range. Obviously the ability to monitor in real-time is lost if the motes are out of range; the design only ensures an accurate reconstruction in case this happens. The motes store locally all SIMs that would otherwise be sent to the base. Since a time stamp is a part of the message, reconstructing the activities while the user is away is done in the exact same way as processing the data in real-time. Because each packet is only 4 bytes, the motes are fully capable of storing a very large number of them before *running* out of memory.

Packet Loss

The base is able to detect any lost packet by examining the packet numbers. If the packet just received is not in sequential order with respect to the last packet received from the same mote, the base signals the mote to resent the missing packet along with the next packet it sends, at which time the base is able to retroactively correct any errors that may be caused by this piece of missing information. Given that the packet number is represented by 5 bits, there would be no problem unless more than $2^5 = 32$ consecutive packets are lost, which is unlikely during normal usage. Before the signal becomes bad enough for this loss rate, the RSSI should be low enough to activate the out-of-range protocol.

3.3 Security

This design does not have any embedded security measures for the transmission of data between the body motes and the base mote. However, there has been research on the establishment of secure connections [15] [16], and the security measures have a very small overhead. A simple symmetric key scheme can be applied on top of the existing protocol for added security in an environment where such measures are deemed necessary.

A user may take off the sensors at certain times, such as while working out in the gym. Then someone else may forge data mistakenly or maliciously by putting on the sensors, performing a sequence of motions, and then returning the sensors to where they were. To avoid this, we can use a simple gesture-controlled security scheme to protect against common attacks. For instance, the user may perform a particular sequence of motions, which locks the sensors, and the processing station turns off its

own recording ability upon this request. The user's actions will be monitored and detected as usual, but nothing is recorded until the user performs another particular sequence of motions which serves as a request to unlock the system. This way, if the user locks the sensors before taking them off, no one can fabricate data into the central station to record unless he knows the sequence of motions required to unlock the sensors. Of course, more traditional measures, which may require additional hardware such as password and fingerprint protection, can be embedded as well.

3.4 Prioritized Motion Data Delivery

To further conserve power without losing the ability to provide real-time monitoring of certain high priority motions, the body sensors can asynchronously deliver data with urgent signatures in real time, while sending less important data at a lower frequency, or even drop irrelevant data points altogether. For instance, while the central processor might not be interested in identifying the details of a half-hour drive in real time, it would want to identify collision take appropriate and time-sensitive actions if one were to occur.

To do so, the body sensors need to be calibrated with a priority table — in which each (x, y) acceleration tuple for a particular sensor has a particular priority. The model identifies the key signatures for high priority motions and informs the sensors which tuples to look for. While each sensor will delay and combine its low-priority messages before delivering to the central server, they all will process and deliver data with high priority motion signatures immediately. If the central processor receives multiple "urgent" messages suggesting the same state, it can send inquiries to the sensors that did not send similar "urgent" messages and obtain their states. From this information, the model determines the motion of the user, and reacts accordingly.

4 Experimental Results

In order to evaluate the performance and reliability of the proposed system, we have implemented it with 6 Tmote Invent sensors, one of which is used as the base mote, while the other five are attached on the chest, both wrists and both ankles (see figure 4). Tmote Invent recharges through its USB connection, and is equipped with accelerometer in both x and y directions. However, this is by no mean the sole platform on which the protocol is practical; in fact, many of the functionalities of this

mote are unnecessary and disabled, such as the built-in microphone and speaker. A smaller, less powerful mote without the extra sensors would be an even better fit for the purpose of this system.

4.1 Metrics and Methodology

In this implementation, we used three metrics to test and verify our result. First, we used the training data obtained from person A to examine the training data itself, in order to check for self-consistency. Second, we tested another set of motions, performed by the same person, to see how well the trained model could identify the actions of the same user. Last, we used a different person, B, to perform a sequence of actions, and asked the model to identify the motions while still trained with only person A's data to see if the model could provide accurate identification. Note here person A and B have roughly similar characteristics (college-age male students), but different weights, heights and other physical considerations. The first test serves as a sanity-test; the second serves as the narrower usage of the training and a base for comparison, and the third test is the most important one in terms of evaluating the effectiveness and applicability of using preset measures to determine a user's motion.

4.2 Accuracy

The self-consistency test passed without much ado and is not particularly interesting; results for the second and third tests are summarized below.

	User A		**User B**	
	% called	**% accurate**	**% called**	**% accurate**
Still	100	100	100	100
Jimmy	100	89	100	99
Wave	98	100	100	100
Walk	91	96	99	95
Jog	93	98	87	90
Sprint	88	100	78	95
Overall	95	97	94	96

Accuracy on Person A

The accuracy on person A represents user-specific accuracy. The processed data are presented above. The first column of the table represents the percentage of time when the model identifies what motion person A is carrying out, while the second column shows that, of the times when the model identified a motion, the accuracy of that identification. The motion with the lowest identification rate is sprinting, with only 88 percent, but with 100 percent accuracy. In other words, while person A is sprinting, the model correctly identified the motion as such 88% of the time, while the other 12% of the time the model is unsure of what motion person A is performing. The lowest accuracy occurs for the motion Jimmy legs; 11% of the time, the motion was mistakenly identified as sitting still, because the close resemblance of these two motions. Overall, the model makes an identification 95% of the time, and of these identifications 97% of the time it correctly identifies the motion.

Accuracy on Person B

The accuracy on person B represents the model's accuracy without user-specific training. The processed data are presented in the table above. The third column of the table represents the percentage of time when the model makes a call on the physical motion person B is carrying out, while the second column shows that, of the times when the model identifies a motion, the percent accuracy of the identification. The motion with the lowest identification rate is sprinting, with only 78 percent, but the accuracy remains high at 95%. About 5% of the time, the sprinting motion is mistakenly identified as *jogging*, due to the similarities of the two motions. The lowest accuracy, at 90%, occurs for the motion *jogging*. The model could only predict 87% of the time when that motion was performed. This is the only motion where both accuracy and calling rate is at or below 90%. Overall, both identification rate and accuracy for person B are virtually as good as person A's, with just 1% lower for both values. This slight difference is insignificant because of the test is only done on a small scale. The result significantly outperforms previous designs' accuracy without user-specific training [5].

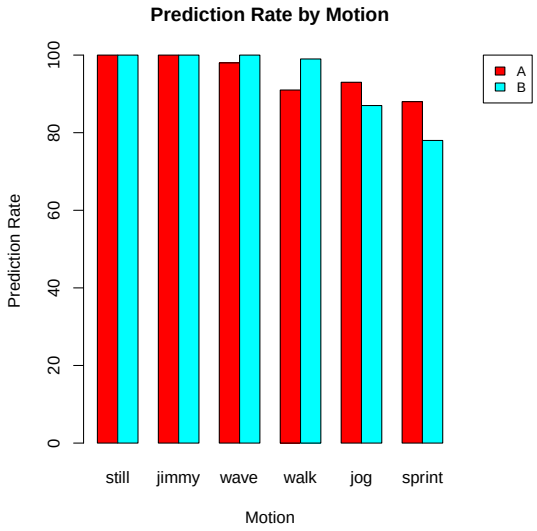


Figure 7: Identification rate comparison by motion, sorted by ascending activeness

Identification Rate Comparison

The identification rates for each user for each motion are summarized in Fig.7. The identification rate is high overall, but the it drops as the motions become more physically active. There is no clear indication that the model identifies user B's motions at a lesser rate than that of user A. The calling rate is higher for A than for B for two motions, the two rates are the same for two motions, and the model identifies a motion more often for B than for A for two motions.

Prioritized Delivery

The experimental data show that while it is impossible to accurately identify the motion with partial data, there are signatures for each motion that would suggest it is taking place. For instance, a series of tests on person A and B rolling off the bed suggest that there is a distinct acceleration pattern the feet undergo in the moment of panic right before the fall occurs, and the chest shows a sudden burst of acceleration upon impact unparalleled by any other motions tested. Because certain signatures are necessary conditions for a motion to occur, it naturally follows that using these signatures as cues to show whether a particularly interesting motion has occurred are reasonable and effective.

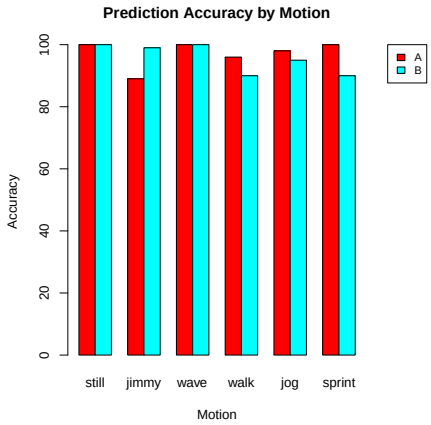


Figure 8: The accuracy of identification by motion for the two users is summarized in this chart. The accuracy is very high overall, and there is no clear indication that the model identifies the motions performed by person B any worse than that of user A.

Further Observation

The model identified the motions *still*, *jimmy leg* and *wave* with virtually 100% of the time with high accuracy for person B, which matched, and sometimes outperformed, similar parameters for person A. In fact, for the motion Jimmy Leg, the accuracy increased significantly, from 89% to 99%. While minor fluctuations would be within the margin of experimental error, there seems to be a trend in the accuracies. This suggests that for some motions, training does not necessarily help reduce error. Rather, the more important factor is people's individual physical habits. In this case, after re-examining the *jimmy leg* habits of person A and B, it is found that while A likes to shake his leg at a low frequency and low magnitude, person B does the same motion with more much vigor. The sensors, therefore, have an easier time picking person B's motion and differentiating it from sitting still. However, for some other motions, such as *jogging* and sprinting, both accuracy and calling rate for person B is significantly lower than that of person A. This suggests that the user specific training will help the model to better identify these motions. While 78% calling rate and 95% accuracy is still much better than models that are very user-specific, the accuracy may still be undesirable for commercial usage.

Overall, the data suggests that the usefulness of non-user-specific training depends

on the type of motion in question. While some of the less physical motions are generic, high-speed actions tend to differ from person to person. This could be overcome by training the model with a large pool of data from multiple people of the same group, which allows the model to focus on the generic characteristics of such motions rather than individual habits. Mathematically speaking, a larger pool with more diverse data would cause the penalization for discrepancy from the mean in the acceleration of certain body parts to decrease, making the model more tolerant of personal habits. We could expect the results for *jogging* and sprinting for user B to be more accurate if B's data were incorporated into the training data set.If we were to pool a sufficiently large number of people from the group, we would expect the accuracy and calling rate of all members in this group (include those who were pooled and those who were not) to converge to some accuracy close to that of person A, i.e. 88% calling rate and 100% accuracy for sprinting, 93% calling rate and 98% accuracy for *jogging*. Nonetheless, even at 78% calling rate and 95% accuracy, the results are fairly promising.

Fig.8 shows the accuracy of the various motions tested for both users across all six predefined motions. We also expect the model to perform better with more fine-tuned parameters. For example, although there already is some bias built into the model to favor the continuation of a motion, the experimental data show that the model still sometimes briefly misidentifies a motion as another, then quickly corrects itself. This suggests that we should increase the bias for continuity of the current motion, and therefore further reduce these types of brief interruptions.

Identification of User B's Walk		
Identified Motion	**Time Stamp (seconds)**	**Duration (Seconds)**
walking	1248720336.4	2.7
jogging	1248720339.1	0.9
walking	1248720340	17.8
jogging	1248720357.8	1.2
walking	1248720359	6.9
jogging	1248720365.9	1.6
walking	1248720367.5	1.2
unknown	1248720368.7	0.3
walking	1248720369	8.9
jogging	1248720377.9	0.5
walking	1248720378.4	3.1
unknown	1248720381.5	0.8
walking	1248720382.3	42.6
unknown	1248720424.9	-

This phenomenon can be greatly reduced if we increase the bias towards motion continuity in the numerical analysis of the current state. Other similar numerical improvements exist and may require more testing before the best parameters can be identified. The table "identification of user B's walk" shows an example of how the system would briefly wrongly classify *walking* as *jogging*, or *undefined*, before quickly returning to the correct identification.

The data for *jogging* shows a similar pattern. The information is summarized in 9. It shows the model's identification over a continuous period of *jogging*. Overall the model correctly identifies the motion, but for three very short intervals the model misclassified the motion of sprinting, and sporadically fails to make an identification for a very brief (less than 0.5 seconds) periods of time.

5 Problems/Limitations

While the design is efficient and accurate, and capable of real-time, non user-specific motion detection and identification, there are a number of limitations and shortcomings. They are

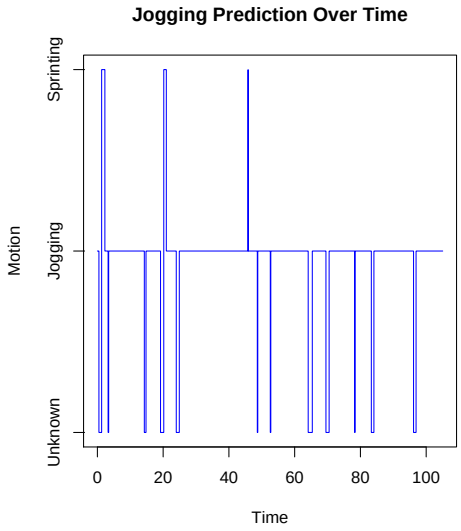


Figure 9: The model's identification based on real time data from the sensors on person B during a 100-second jog.

1. For simplicity, the experimental data collected is based on a 5-motion system, and anything beyond one of those five predefined motions are simply classified as *unknown*. In reality, there are many more motions interesting to a user. More trials with more motions should be carried out to verify the results, and to test the expandability of the design.

2. While it is possible to easily incorporate dozens of new motions into the current pool of motions, there is a limit on the total number of motions the current design could handle. As more motions are added, the resolution of the measurements might have to increase, which will require additional bits to hold and represent the data.

3. Even though most people should fall into particular categories and therefore be able to use preset values for motion identification, outliers', disabled persons' and professional athletes' motion signature may drastically differ from all of the presets and may require personalized training of the system before accurate motion detection. Those individuals' unique gifts and/or conditions may make it impossible for them to take advantage of the generic presets.

4. The issue of privacy and security is not extensively addressed here in this chap-

ter. Because motion identification deals with sensitive personal information, adequate security measure, most likely using shared symmetric secret key [16][15], must be used in a security protocol designed to defend against a wide array of malicious activities.

6 Summary

Through the design, implementation and experiment, we have demonstrated that the task of motion detection and identification can be achieved using a small cluster of sensors and low-resolution data values. Without losing accuracy, the design can be much more efficient than previously thought possible by deploying a series of tactics and by reducing the number and size of the sensory data. We have conducted experimental study on this motion identification design. The experimental data suggest that the proposed design can efficiently, effectively and accurately identify a range of predefined motions, with or without personalized training of the system beforehand. The chapter attempts to present a collection of energy saving techniques to provide a solution satisfying key functions envisioned in the field of body sensor network, while maintaining reliability. The design can support a large-scale implementation where many users can have their own systems, each supporting a large set of motions. Users can opt out of personalized training by using the factory preset values most akin to their own descriptions. The system also allows for personalized trainings without any changes to the motes, making it possible for an average user to do so without any technical background. More sophisticated uses of the system can allow all personal data to be centralized, processed and recorded for military, national security and/or medical uses.

There are several interesting findings. First, the system can help identify motions which different people perform very differently, and those that are similar across the entire population. Anomalies, especially in young children and elderly, may be warning signs of developing physical problems and may require additional diagnosis to see if some form of physical therapy may be necessary [17].

While a resolution of only 3 bit is enough for a limited number of motions to be accurately identified as the chapter has shown, more specialized uses of the system may require finer information. Every time the number of bits used to represent the data is doubled, the number of unique combinations, which is the maximum

number of possible motions, increases by a factor of $2^{10} = 1024$. This allows for easy upscale adjustment of the system to satisfy users' specific needs. The energy-saving strategies make no assumption on the environment the motes are in, allowing the users to come in and out of the range of the base mote, and the design can be integrated into smart environments where a computer can assemble a more complete picture users' actions beyond just their motions, and include information such as location, speed, temperature etc. as well. The array of energy-saving strategies presented, including SIM, local maximum delivery and assumed continuity can be partially or fully incorporated into other designs to maximize performance without losing efficiency.

If further energy reduction is needed, the protocol can run in an even more efficient prioritized mode where only crucial motions are identified in real time without any delay, and some local processing handles when to alert the central model of immediate danger. Most tuples of information will be stored locally and sent out after a longer period of time. This offers enhanced efficiency, and treats different messages with different priority, hence guaranteeing real time response to urgent messages without flooding the base with less important data points.

Part III

Vehicular Sensor Network

To better analyze the process of designing protocols in the second stage of sensor network protocols, we use vehicular network as an example to show how to establish security on communications between weak devices. This chapter highlights the importance and difficulty of the transporting information from local sensors to a central processor.

In this network architecture, it is crucial to provide security support – only authorized vehicles can feed data into sensors and to obtain data from sensors. To block unauthorized and malicious vehicles, data collected by sensors must be encrypted. However, merely encrypting the data cannot prevent a malicious car to obtain the scrambled data. Although the encrypted data is of little use to the malicious vehicle, it is a serious problem when sensors expect vehicles to harvest all the data and carry them to a central station; a malicious vehicle can simply trap the data and leave a hole in the designated data repository. Therefore, authenticating a passing vehicle before transferring any data is indispensable.

A straightforward solution is to use a public-key based scheme, since some (e.g., the ECC) of the schemes can be implemented efficiently on sensor platforms. Taking a closer look at the problem in a real experimental study, we found that authentication takes about one to two seconds in many cases, which is non-negligible for a car traveling at tens meters a second. A car may rush out of a sensor's transmission range after the authentication is conducted. In this chapter, we show our security solution to the vehicular sensor networks and give experimental results on a realistic deployment. We show through a simple secure protocol design and implementation the feasibility of secure data collection in a vehicular sensor networks. We deployed sensors along the road side to test the performance of the communication between the roadside sensors and the sensors in a moving vehicle. We demonstrate the protocol works in a realistic setting by collecting the real trace data through real implementation. We hope this research shows valuable experience in deploying security support for this type of networks.

1 Related Work

A survey of the security of vehicular network can be found in [18]. There are many vulnerabilities for an unsecured network, such as jamming, forgery, impersonation, and in-transit traffic tampering [19]. Research shows that a symmetric key scheme is required [18]. A number of researching teams have put forth many solutions to group key and authentication problems [16, 20, 21, 22, 23]. Some of them utilize the Cabernet system where data is delivered opportunistically during travel. While less powerful, it does provide a quick solution that can be implemented without an overhaul [24] [25].

There have also been some research exploring the possibility of using a certificated-based protocol, such as the one proposed by Wang et al. [26]. It uses the short-range radio communication of motes to pass information from various roadside measuring devices to an information-gathering car mote, which then carries the information to a computer to process. However, it does not address some of the issues unveiled by Balfanz et al. [16].

2 Problem Setting

In this chapter we assume the following environment and the availability of equipments:

- Many stationary sensors deployed on the side of the road that can detect, measure and record a certain aspect of the traffic pattern, such as the speed of vehicles in its range. Such motes are currently commercially available. It does not possess significant computational power, nor does it have much storage space.

- Another mote similar to the stationary ones placed in a car that can gather information from the stationary motes and then deliver it to a computer in the car. Its responsibility requires it to be able to securely communicate with other motes and with a computer.

- An upload server, which is to be located at the end of the road. It should be a computer with Wi-Fi capacity. It will obtain the relevant information from the computer in a car once in range. This upload server can be located in a

toll gate, rest area, or any other similar structures. This server should have the ability to process and analyze the raw data, and notify relevant parties of its findings. As a computer, the upload server has much computational power. It is also assumed that its storage is virtually unlimited. This is justified by the fact it can communicate with external servers across the Internet. However, the upload server cannot communicate directly with the stationary motes.

- An authentication server, which is to be located at the beginning of the road. It should also be a computer with Wi-Fi capacity. It will permit the car mote to communicate with the stationary mote after the server verifies the car mote's identity. The exact protocol of authenticity between the car mote and authentication server is not discussed in this chapter. This server, like the upload server, is assumed to have great computational power with virtually unlimited storage.

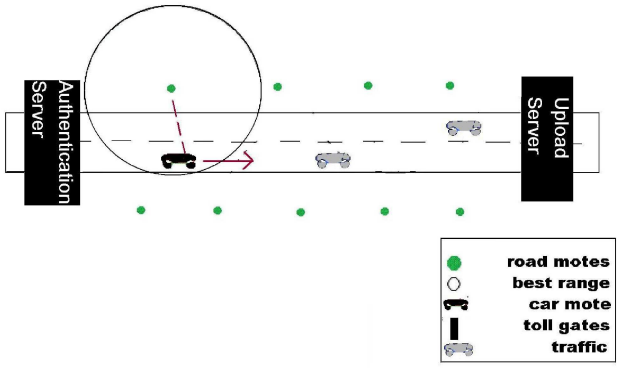


Figure 10: A scenario of collecting data from roadside sensors in a vehicular networks

2.1 Adversary

Our goal is to design a protocol that is secure, reliable, and efficient. In the following we show our assumptions about the adversary.

The adversary is an unauthorized party that wishes to either obtain the secure information gathered by the roadside mote or at least block the car mote from gathering it. The adversary is assumed to have the ability unlimited access to any public information. It may try to impersonate the car mote so steal the data, or it may pretend to be a stationary roadside mote to provide the car mote with falsified data.

Since the car mote is not a part of the infrastructure, it is even reasonable to assume that the car mote is malicious. However, this is not a major concern, since it is assumed that the authentication server will accurately identify any malicious parties before granting access. But in case the adversary successfully obtains the session key, it still cannot forge inaccurate data to the US, for it does not know the secret key.

The adversary is not expected to have the secret key, or know the hash function with which the secret message is encrypted. It is not able to physically damage any of the structures mentioned above, nor is it able to rewrite any pieces of software implemented on either the servers or the motes. That is, the adversary is an outside party without the knowledge of the inner workings of the system. The adversary is also not expected to be able to crack the security via brute force. This assumption is reasonable since exemplary hashes are believed to be computationally infeasible to solve within any reasonable time frame [26].

3 Protocol

The following protocol is designed in order to balance security with efficiency. It provides a 3-key system that safeguards against various malicious parties, as well as a four-way handshake that allows the roadside mote and the info-gathering car mote to mutually authenticate. The symbols used in the protocol are summarized below:

Variable	**Symbol**
Car Mote	C
Roadside Mote	M
The secret massage	m
The authentication server	AS
The upload server	US
The session key generated by *AS*	k
Secret value known to *M*, *AS* and *US*	s
Randomly generated challenge value	r
Hash function of *r* and secret *s*	$hash(r,s)$
Random number used in the last step of verification	r_t

The first of the two servers provides car mote C with the necessary authentication information (denoted AS), while the second uploads and processes the data collected

by C at the end of C's trip (denoted *US*. M wants to transfer data reliably to C as a car passes by. On the road, C will mutually authenticate with M and collect data encrypted with the secret key. At the end of the road, C will send all collected data to the upload server, which can use the secret key to decrypted the messages.

3.1 Pre-distribution

The servers share the secret key s with M. No one else knows this key.

3.2 Communication between *AS* and *C*:

To provide C the ability to access the information M holds, AS generates a random number R, and use this R to form a session key k:

$$k = hash(R, s) \tag{4}$$

AS performs this because it does not want C to know s, in order to prevent malicious C to forge fake sensors. After this process, AS gives k and R to C securely. C will now possess all the information it needs to collect data from M.

3.3 Communication between *C* and *M*

C broadcasts probe messages containing R on the road. When C and M are within the range of communication, M would receive the message and generate the session key $k = \text{hash}(s,R)$, which is the same key known to C. M can then perform a 4-way handshake with C:

1. M generates a random challenge r, and send it back to C;
2. C generates another random number R_c; compute a temporary key;

$$TK = hash(k, r, R_c) \tag{5}$$

   then it generates a MAC for R_c by using this TK; then sends both of them back to M

3. M compute the TK in the same way and use the key to verify the MAC. M can then send back another random number r_t with the MAC generated by using this TK. After verification, C will confirm that M get the TK correctly.

4. After that, both motes are authenticated with each other. M sends a success message and both sides can then start encrypted data communication.

3.4 Communication between C and the US

Upon arriving in the range of the upload server US, C uploads all gathered data as well as the session key to it for processing. US should have the ability to decrypt the encrypted information with the secret key, and it can verify with the AS that the session key is indeed a valid one. Once uploaded, C has finished its mission and can now reset.

3.5 Summary of the protocol

The following summarizes the protocol outlined above.

AS to C	:	R, k = hash(R,s)
C to M	:	R
M computes	:	k = hash(R,s), generate r
M to C	:	r
C computes	:	generate another random number R_c
	:	TK = hash(k,r,R_c)
	:	generate a MAC for R_c using TK
C to M	:	MAC, R_c
M computes	:	TK in the same way; verify MAC, generate r_t
M to C	:	MAC, r_t
C verifies	:	MAC; handshake complete

Security Analysis

The protocol utilizes a four-way handshake instead of a shorter one because of the complexity of the structure. In order to have optimized security, three distinctive key types exist in the protocol. They are the secret key (which only roadside motes M and the servers know about), a session key (generated by AS and held by C) and the temporary key (for the one time use between C and a M). The handshake protocol allows C to authenticate M, because M needs to be able to generate the same session key C holds with the provided random number. This is impossible to

do without the knowledge of the secret key, so no adversary can forge sensors to provide falsified data to the car mote C. At the same time, the handshake grants M an opportunity to authenticate C. C is required to compute a temporary key using the random number M provided, and it is required to generate a MAC using this key. The value of the MAC is verified by M, and a right value means the session key is indeed valid. This would block malicious vehicles from impersonating $C$to obtain crucial information. Moreover, to protect the secret key, the session key exists so that C will not have the most fundamental knowledge of the architecture. It cannot falsify information by faking data from non-existing roadside motes either. Therefore, the protocol provides defense mechanisms against malicious car motes, forged roadside motes and impersonation of the car mote.

In this chapter, we merely consider a trusted vehicle to carry the data to the upload center. In reality, the vehicle may drop the data after collect the data from the sensors. In that case, we can use a simple strategy to circumvent this behavior. The idea is to let the upload server to notify the authenticate server what data has been uploaded successfully at the upload server. When another vehicle comes to offer assistance in carrying the data, the authentication server will pass the already collected data in a bitmap along with the authentication keys to the vehicle. When the vehicle moves along the road, it can notify the sensors the data that has been collected (the sensor will have to check the validity of the bitmap for the collected data) so that the sensors can proceed to delete the data from its memory since it is certain the data has been dumped on the upload server.

4 Experimental Results

To evaluate the proposed protocol, we have implemented it on two TelosB motes, one of them is used as M, while the other C. TelosB is powered by the MSP430 microcontroller. MSP430 incorporates an 8MHz, 16-bit RISC CPU, 48K bytes flash memory and 10K RAM. The RF transceiver on TelosB is IEEE 802.15.4/ZigBee compliant, and can have 250kbps data rate. While the hardware directly affects the RSSI and other aspects of the experimental results, TelosB is by no mean the sole platform for the protocol is practical.

4.1 Metrics and Methodology

In this implementation, we used the following three metrics to better evaluate our results: received/dropped packets, received signal strength indication (RSSI), and the displacement across which the packets are transferred. On an open stretch of road, we drove past the roadside mote M at different speeds with the car mote C on top of the car and connected to a laptop (via USB), which runs a Java program that reads, records and analyzes the data received. M continually sent out radio transmissions in an infinite loop. Upon entering M's range, C picked up the encrypted message *hash(r,s)*, decrypted it and responded, and finally obtained the message m.

4.2 Mote to Mote Communication

Three distinct tests prove conducted. C is driven by M at the constant speed of 30, 50, and 70 km/h. Three trials were conducted for each speed. In addition, stationary tests at fixed displacements before and after M along the road were conducted at the displacements of 25, 75, 125, 175 and 225 meters. A unique ID number was assigned to every packet received for easier identification. The ID, the RSSI as well as the packet's time of arrive (an offset from the start time) were recorded for each trial. Also recorded was C's approximate displacement from M. From this, it was possible to calculate when, where, how many, and at what speed packets were dropped. The location of C along the road is expressed in terms of its displacement from M, where a negative value X represents that C is X meters away from reaching the same point along the road as M, a value of 0 represent that it is at the exact same point along the road, and a positive value Y represent that C has passed M by Y meters.

In order to better analyze the relationship between possibility of dropping a package and the distance between the two motes, in Fig.11, we graphed the total number of packets received/dropped for the three trials at 30km/h. From the graph, it is evident that at the possible range for C to receive packets from M is around between -150 meters and 50 meters. This represents a window of opportunity of about 200 meters, or about 24 seconds. However, significant number of packets was dropped from -155 meters to about -55 meters, and again resumes to drop significantly at around 25 meters. Therefore, the most reliable window of communication where very few packets (less than 5 percent) is around -55 meter to 25 meters. This 80 meter window represent about 10 seconds.

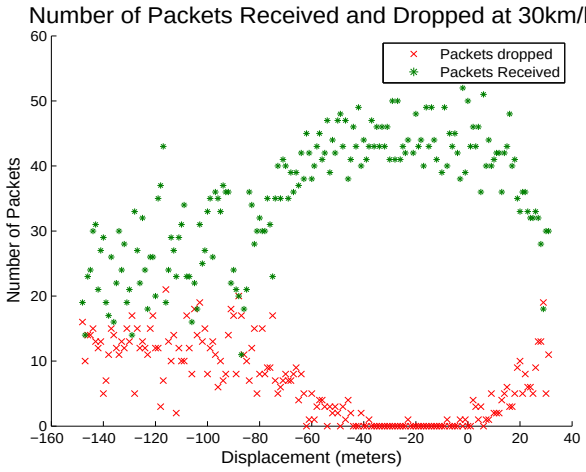


Figure 11: This graph shows the number of packets received and dropped between the -150m and 50m, and it is the total of three separate trials. It shows that from about -40m to 0m, virtually all packets are successfully received for all three trials. It also shows that close to half of the packets are dropped near the ends of the graph

The data of 50 km/h and 70 km/h show a similar pattern. However, the optimal window of transmission is halved to just under 6 seconds. This could be too short to fully and securely transfer all data based on the currently protocol. The possible range of reception, optimal range of reception and approximate time frame to transfer data within the two ranges are summarized in the table below:

Analysis of Possible/Best Packet Transmission Frames

Car Speed	**Possible Range**	**Optimal Range**	**Possible Duration**	**Optimal Duration**
km/h	meters	meters	seconds	seconds
30	[-150, 50]	[-55, 25]	24	10
50	[-150, 50]	[-75, 5]	14	5.8
70	[-150, 50]	[-85, 25]	10	5.7

The table is populated with the average of the three trials for each speed. Possible range represents all displacement values where transmission is possible, whereas the optimal range represent the best range for transmission as discussed above. We can use our knowledge of the best window of communication to increase the reliability of our protocol. Possible/optimal duration represents the amount of time in seconds C will stay in the respective ranges.

4.3 RSSI

The signal strength (RSSI) is plotted against displacement in Fig. 12 to better analyze the relationship between the two. Three trials show an identical trend with slight horizontal displacement. The signal strength for the first trial peaked at about -13 meters, second trial at -5 meters, while the third one peaked at -26 meters. It is, however, clear from the graph that RSSI increases as the displacement narrows. The fact all three trials peaked at slightly negative displacement suggests that the radio signals are stronger before C passes M. However, there is little difference between about -175 meters and -75 meters in terms of RSSI, suggesting that the strength is not simply inversely proportional to the displacement. This information provides insight to the best timing of the handshake process, and we can use the RSSI as an indicator to show if the communication strength is high enough for the security protocol to start. At 50km/h and 70km/h, a similar pattern is shown. The results are summarized below:

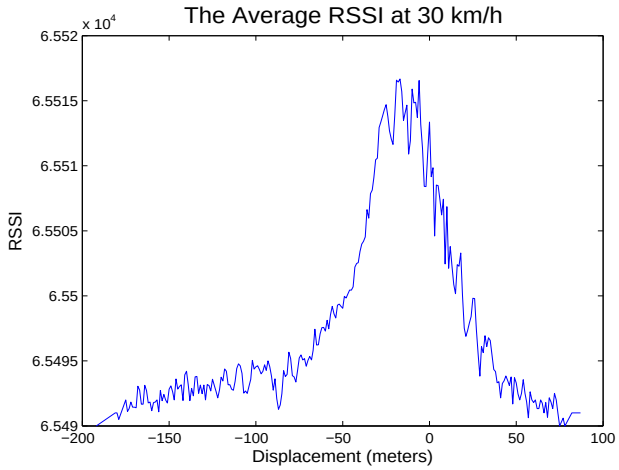


Figure 12: The RSSI of the packets received within [-200m, 80m]. At about 50m away from the roadside mote the transmission becomes reliable, until it passes the roadside mote by about 25 meters. It peaks shortly before reaching 0m.

Approx. Displacement of Best RSSI

Car Speed	Displacement	Time
30 km/h	-8 meters	-1.0 sec
50 km/h	-16 meters	-1.2 sec
70 km/h	-23 meters	-1.2 sec

The trend shows that as speed of the vehicle hosting C increases, the ideal displacement from M for packet transmission becomes increasingly negative. The ideal amount of time for packet transmission, on the other hand, seems to be about just more than 1.0 seconds regardless of the difference in speed.

4.4 Security

The amount of time authentication of this protocol takes is compared with other schemes in order to better analyze the efficiency and dependability of the protocol. In particular, the amount of time needed to encrypt the message with AES takes less than 1ms with 16 byte-long keys and random numbers. SHA-1 would take 4ms. The proposed protocol can use either one of those two. The speed of encryption/decryption is extremely important in the context. This is because the car mote only has about 6 seconds to communicate with a roadside mote at high speed, and we must ensure not only that we have enough time for the handshake, but also that the security part takes only a small fraction of the amount of time we have, and that sufficient amount of time is left for the actual data to pass through. On the same platform, we found that the ECC-based encryption needs 2 point multiplications, which takes roughly 3.1 seconds on the TelosB. Decryption will take one point multiplication, or about 1.55 seconds. This suggests that an asymmetric key scheme would take much longer to establish the secured connection and is not be suitable for a similar set up. Using the symmetric key protocol as proposed, the total amount of time needed for the motes to encrypt/decrypt is negligible. In addition to the calculation time, however, we also need to consider the amount of time it takes to actually transmit handshake messages. The protocol requires a 4-way handshake, 2 packets to be sent from the car mote to the roadside mote, and 2 more going the other way. At the experimental rate of 13 packets per second on average, the communication will take roughly 0.31s. Therefore, the total amount of time required for the authentication is about one third of a second, or less than 5 percent of the total amount of available time inside the optimal transmission range. This is very acceptable.

4.5 Further Observation

At high speed, there would be about 6.5 seconds left for the transferring of interesting data. The TelosB hardware has a maximum transfer rate of 250kbps (as declared on its specifications) and an experimental rate of 200kbps at 70km/h. This means that it will take about 6 cars to unload all 1024K bytes of data the hardware can hold at a time. However, in a realistic setting, we should extract the data long before the on board flash memory is full. The low percentage of communication time shows the protocol is useful in a realistic setting. In the experiment we did not try to tweak the packet size to maximize the amount of data transferred at a time; ideally, we should be able to reach the rate of 250kbps, and only five car motes need to drive by to collect all data resting on the roadside mote.

5 Summary

In this chapter, we show our design of a secure data collection protocol for vehicular sensor networks. We conducted experimental study on this protocol. There are many interesting findings in this implementation. First of all, it shows that many of the goals of vehicular network communication can be easily achievable. At a speed as high as 70 km/h, the motes still have about 5.7 seconds of optimal transmission time, which should suffice under normal circumstances. Another interesting conclusion drawn from the data is that the signal strength is better when C is still some distance away from M than when the two are at the same point along the road. This is confirmed by the fact the optimal range of transmission is not centered at a displacement of 0 meters, but at around -30 to -15 meters.

Part IV

Location Proof via RFID

Using RFID tags as the basic hardware, this chapter presents a reduction in computational intensity and hardware sophistication required to provide location proof service. We hope that the solution to this real life application will also provide insight into a posteriori security measures, and the central server's role in consolidating a large amount of information from local sensors. It highlights the importance of the third and last stage of sensor network protocol, central processing.

Since the invention of RFID in 1948, the technology has gone a long way to become an integral part of our society[27]. More and more expensive hardware are being replaced by these smart tags, suggesting that the broad usage of this cheap medium is becoming commonplace. One area we believe RFID tags have potential in is the field of location proof service, which seeks to provide a means for clients to show that they are present in a particular place at a particular time.

It is important to keep consumers in mind and seek to provide realistic solutions, and "the difference between successful and shunned RFID applications turns on delivery of clear, tangible value to the average consumer[28]." In this chapter, we present a protocol and a solution that can adapt to different situations with respect to real life needs. Possible applications include replacing some of the more complicated and expensive systems in place today, providing location-based security measures, and deploying location proof services to applications previously economically prohibitive.

The key challenge is that the smart tags, unlike more advanced sensors, do not have the energy required to keep their own reliable clock. Accurate timekeeping is a necessity for location proof service. While such an obstacle can be overcome by making a server or other powerful computational devices an integral part of the protocol, such systems would not be qualitatively simpler or cheaper than existing technologies. Real-time server interaction may also be impossible for certain applications. For instance, although we may wish to provide location proof service for subway systems in metropolitan areas, it is often the case that cell phones have an unreliably low signal. While the cell phones can communicate with the local RFID tags, real-time interactions with servers become impossible in this situation. This chapter proposes a design that can provide location service proof without real time aid from more

advanced devices, broadening potential applications while reducing the cost.

The design we introduce here is lightweight, scalable and parametrically adaptive, and it can provide a quantitative guarantee on the accuracy and performance of location proof service using RFID tags. This is the first paper to our knowledge that approaches this problem from such a perspective. Although at times the system is less precise than some of the other designs using motes already proposed in the academic community, the flexibility and ultra low cost makes it a very competitive approach.

This chapter first introduces a straightforward protocol for providing location proof service without the need for real-time communication with the server, then provides support for the protocol with several iterations of mathematical solutions that are progressively more advanced and capable in stopping the adversary to increase the quality of service, and lastly examines the proposed solution in terms of performance and accuracy.

1 Related Work

There has been a lot of work done in the field of location proof service, primarily designed and implemented using sensors, computers, or other similar advanced hardware [29, 30, 31]. While they provide a cheaper and easier means to provide location proof than camera-based conventional methods, the cost of deployment and maintenance is still substantial. The design presented in this chapter attempts to use RFID technology, which is significantly cheaper than other electronic devices, to achieve the same end.

The low computational and storage capacities of RFID tags compared with sensors and computers pose a challenge. Previous results such as [32, 33, 34] provide important insight into designing systems and protocols that help increase the reliability of the location proof service and reduce the malicious party's means to obtain location proof falsely, but they tend to bypass the challenging issue of not having a reliable clock by limiting the application of the design or by real-time interaction with more sophisticated hardware. Without an intelligent design that derives the actual time, the protocols must rely on external trusted sources such as an online server to provide the certified time.

Another area many researchers have focused on is the issue of privacy. The pervasive use of RFID technology has raised many concerns because users can give away

personal information unknowingly. The extensive research in this tries to identify and address this concern [18, 35, 36, 37]. This chapter is designed to provide voluntary location proof service, not monitoring service, so concerns about privacy will not be a focus of this thesis.

2 Problem Formulation

2.1 Setting

The protocol and solution proposed in this chapter attempts to describe a means of efficiently and inexpensively providing location proof service using RFID technology. The applicable situation has the following characteristics:

1. The service is used frequently and by many users, and
2. Minor inexactness in the time of visit is allowed

The entire process requires little or no user interaction and can be fully automated. The design is a good fit for most situations location proof service may be required, at established public locations such as shopping centers, public transportation, parking garages, offices, restaurants and hospitals. Potential uses of the design can provide service at a much lower cost than contemporary measures and can help extend location proof service into sectors previously prohibited due to economical and technical concerns.

2.2 Goals

We wish to provide location proof service seamlessly, accurately, and efficiently to our clients.

1. *Seamless* The design should require minimal effort on the clients' part to set up and use, and should be able to conduct the protocol completely automatically if the clients so choose.
2. *Accurate* The time of presence should be correct within an allowable margin of error predefined depending on the application. In general, we can define the time of presence as correct if the recorded time is within the fluctuation range

of good users' clock. In other words, provided that a user is not experiencing technical difficulties in obtaining the time, and he is not deliberately falsifying the time to the system, we can define the time given as correct.

3. *Efficient* The calculations tags and readers must compute should be trivial because of hardware and power management considerations. The calculations on the server side can be much more intense, but should still be reasonable with respect to the amount of data, the level of accuracy and other parameters. We envision a sizable system to perform daily calculations that should finish within a matter of minutes.

2.3 Adversary Model

Based on common assumptions, we define the malicious party to have the following goals, capacities and limitations.

Goals

The malicious party has two primary goals:

1. Falsely obtain location proof, or
2. Disrupt the service by making the solution wrongly record the time of presence for other users

To achieve the first goal, the malicious party may either falsely report the time at which he is present at the particular location, or not to be present in the location at all. For example, if the malicious party wants to demonstrate falsely that he is present at the opening ceremony of the Olympics, he can try to either show up at the Olympic stadium before or after the ceremony and obtain a location proof with a false time stamp, or not to show up in the Olympic stadium at all but still pretends to do so.

Capacities

The malicious party is expected to do the following:

1. Insert an arbitrarily large number of data points designed to dilute and contaminate the data pool in order to either invalidate legitimate users' claims or corroborate malicious users' claims.

2. Perfectly coordinate all the malicious users' actions towards a common goal

3. Eavesdrop on legitimate users' communications

Of the three, the first one is by far most profound and powerful. It means that it is within the malicious party's power to submit as many data points as it is physically feasible, and for any given pool of data points, a majority of them could be potentially malicious. While it is reasonable to argue that for certain applications the malicious party would not have enough access to dominate the data set, making an assumption on the upper bound of malicious data input is difficult and unreasonable for many other situations. This also implies that any successful countermeasure must rely on the consensus of unique users, not of the data points.

Limitations

Obviously the adversary cannot be omnipotent, and one of the fundamental strategies deployed to safeguard the protocol is to identify potential malicious party's weaknesses, and utilize them to our advantage. The key limiting factors on the malicious party are:

1. Because of the computational intensity required to crack even short secret values via brute force, and the secret values built into the RFID tags are known to the central server only, it is reasonable to assume that the malicious party cannot compromise these values.

2. Because of the level of conspiracy and organization required to amass a large number of malicious users to systematically deliver falsified data to the server, we assume that there is a tight upper bound on the number of malicious users present at the same time.

3 Protocol

This section describes a protocol that does not rely on real-time server interaction with the user, where the server can be contacted at some time after the location proof handshake with the RFID tag for the users to obtain their location proof.

3.1 Symbols

Variable	Symbol
Reader	R
Server	S
time	t,$\hat{t}$
random number challenge	r
Tag's unique names	T_{id}
Secret values known to T_{id} and S	S_{id}

3.2 Overview of Protocol

The protocol contains two major components. The first component is exclusively between the reader and the tags, carried out locally in real time. The second component can be carried out at a later time, and is exclusively between the reader and the server. In the first part the reader sends the time to the tag and the tag stamps time submitted, and provides the reader with all the information it needs upload to the server later. In the second part, the reader uploads all relevant information to the server for verification, and the server processes the information by comparing it with all the other location proof service requests. The server will consolidate the information from all the users to determine the accuracy of the readers' claims, finally inferring the actual time of presence from the data available. The location proof is then sent back to the reader.

3.3 Protocol Details

Pre-distribution Arbitrarily many RFID tags can be distributed for the purpose of providing location proof. Each should hold their own unique ID and secret value, and all such ID-value pairs should be known to the server and no one else.

Communication from Reader to Tag Because the tag has no other ways of obtaining the time, the hello message from R to T_{id} should contain the current time. The tag processes the time by computing the encrypted value v and increment its internal counter, where

$$v = h(t, n, S_{id}) \tag{6}$$

t is the time, n is the current count (i.e. the sequence number of the data point), and S_{id} is its secret value.

Communication from Tag to Reader The tag then sends v, n and its ID back to the reader.

Communication with the Server The user can hold on to the information provided by the tag, without any time-sensitive need to upload any information to the server. However, when the user is ready to obtain the location proof, it can carry out the following.

1. Sign the T_{id} for later verification purposes, and
2. Upload the signature, v, n, T_{id}, t and the public key to the server.

Upon receiving the location proof service request and all the aforementioned material, the server verifies the following information before providing the location proof to the user:

1. The user's identity: First look up the user in the database to make sure the user is not using multiple keys to pretend to be multiple users, then check whether $R_{public}(signed_R(T_{id})) = T_{id}$ and determine if the user is who he claims to be.
2. The place: Carry out the the same operation as the tag did in equation (1), and verify that the result matches the value v provided.
3. The time: After checking the validity of the data point, the server should add it to the solution (described in detail in the following section) to calculate $\hat{t}$, the adjusted time.

Once the server obtains all three pieces of information, it can provide the location proof by

$$\texttt{location proof} = signed_S(signed_R(T_{id}), \hat{t}) \tag{7}$$

which summarily embeds the who, where and when.

Lastly, the server sends the user the above location proof, $\hat{t}$ and the T_{id}, where the latter two serve as a readable description of the content of the location proof.

3.4 Protocol Summary

The protocol designed to provide location proof with one server and an arbitrary number of tags, and no real-time interaction with the server is summarized below.

R to T_{id}	: t
T_{id} computes	: v=h(t,n,S_{id}), $n++$
T_{id} to R	: v,n,T_{id}
R to S	: $signed_R(T_{id})$,v,n,T_{id},R_{public},t
S verifies	: $R_{public}(signed_R(T_{id})) = T_{id}$, v=h(t,n,$S_{id}$), calc $\hat{t}$ from solution
S to R	: $signed_S(signed_R(T_{id}), \hat{t})$, $\hat{t}$, T_{id}

3.5 Protocol Discussion

The protocol helps defend against many malicious attacks, as well as ensure no personal or crucial information is passed back and forth when the reader communicates with the tags; therefore, an eavesdropper cannot hope to capture the radio signal and harvest users' private information. There is also little setup, and clients can be added and removed from the system seamlessly. A malicious user cannot hope to obtain a location proof without being physically present at the location.

On the other hand, the protocol relies heavily on the solutions from central processing and consolidation of data points. The malicious party can falsely obtain location proof and can disrupt the system by directly injecting data points into the central processor, submitting data points the same way as a good user. In particular, a malicious user can report falsely the time of presence, hoping to obtain a location proof with a time stamp different from when he is actually there. He may also report a large number of data points to support his false claims and distort others' time of presence. This translates to the server's responsibility to sort out a large number of data points.

4 Solutions

4.1 Mathematical Problem Formulation

In order to achieve the general goals outlined in the problem formulation section, we must define the optimization problem mathematically. In general, we wish to find a monotonously non-decreasing series $t_1 \leq t_2 \leq \cdots \leq t_n$ that best describes the behavior of the data we collected. From the indices of the data points, we know the chronological order the data points should be in. However, there is no guarantee that the time reported in those data points would correspond to the same order. In case a data point has a larger index but an earlier time stamp, the central server must resolve the discrepancy. In other words, we need to derive the a series subject to the monotonicity constraint that best resembles the data points reported in individual location proof service requests.

The first and foremost difficulty is to define the problem rigorously. We present three solutions. They directly correspond to three different definitions of the optimization functions, and they are progressively more complex and better capture the end-goal of the protocol.

4.2 First solution: ℓ^1 isotonic regression

We can define the problem as deriving a monotonically non-decreasing series that minimizes the ℓ^1 mismatch. This solution would treat all data points equally and attempt to find the time series that best resembles the data given. Since it seems impossible to tell malicious and good data apart, this solution seems to be the best we can do given how little control we have over the data. A ℓ^1 fit is chosen rather than ℓ^2 or ℓ^∞ because ℓ^1 is the least sensitive to outliers, so ideally sporadic data points with incorrect values would have small impact on the final form of the solution.

Formally, we can define the problem as that of finding $t_1, \ldots, t_n$ that solve

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n} |\, t_i - d_i \,| \\ &\text{subject to} && t_1 \leq t_2 \leq \cdots \leq t_n. \end{aligned} \tag{8}$$

The advantage of the linear programming formulation Eq.(8) is that linear programs can be solved very efficiently.

We do not assume a bound on how many bad timestamps the malicious party can inject into the data pool, and there is no restriction on how far from the actual time the malicious data can be. As a consequence, it is not hard to see how the malicious party can take advantage of the situation. With a large number of misleading information, the malicious party can manipulate the final solution however they like. The worst-case effect that the malicious party can achieve is given by the optimal solution of the dual of the convex program Eq. (8):

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n-1}(d_i - d_{i+1})y_i \\ \text{subject to} \quad & y_1, \ldots, y_{n-1} \geq 0, \\ & y_1 \leq 1 \\ & |\, y_2 - y_1 \,| \leq 1 \\ & |\, y_3 - y_2 \,| \leq 1 \\ & \vdots \\ & |\, y_{n-1} - y_{n-2} \,| \leq 1 \\ & y_{n-1} \leq 1. \end{aligned} \tag{9}$$

The dual suggests two approaches a malicious party might take in order to influence the ℓ^1 isotonic regression.

An obvious approach would be to introduce a large number of consecutive timestamps that are monotonically decreasing (i.e., a long string of timestamps for which $d_i > d_{i+1} > \cdots > d_{i+N}$). This attack is ineffective, however. If a single individual attempts to do this then that individual's timestamps will run backwards in time and will be easily detected. Similarly, if a small group attempts to produce such a string they will be detected; they are still limited by the number of the group if they wish to escape detection. Since we assume that the malicious party cannot marshal large number of malefactors, this approach is very limited.

A second approach would be to introduce self-consistent timestamps that are out of sync with legitimate timestamps. That is, the malicious party would interpolate false timestamps amongst the true timestamps. In this attack the timestamps of any single user would be monotonically increasing and thus self-consistent.

4.3 Second solution: greedy removal of the biggest offenders

Motivation The second approach the malicious user can utilize as described above creates a major challenge to the first solution, because the strategy makes it

difficult to discriminate between honest and malicious users. Minor modifications of ℓ^1 regression do not address this problem, so we must have some way to filter out the malicious users in order to create a good solution.

Description In response to the challenges to the first solution in the discussion above, we propose a second solution that seeks to identify the outliers and remove them altogether from the data pool beforehand. We only care about the total inconsistencies between the remaining data points and the monotonous time series. More specifically, it will attempt to address the issue by searching through all the users upon completion of the aforementioned solution selection technique, then identify the users whose data points poorly fit with the final solution. That is to say, identify and remove all the data from the user whose removal will significantly reduce the residual of the ℓ^1 monotonously non-decreasing fit. There are three ways to view which user is the greatest offender, per capita, per data point, or somewhere in between. The first will aggregate the total misfit of all the data points from a single user, so the removal of that that user will result in the greatest drop in the total residual. The second way will scale each user's contribution to the misfit by the number of points he provides, and the last way will scale users contribution to the residual by some other number, seeking a middle way between the first two methods.

Comparison of the three methods The problem with the first method of using total mismatch is that it harshly punishes the users who provide many data points, regardless of their intentions. The malicious party can evade detection by submitting a few points that are highly divergent from the true values, but because they malicious users provide a scarce of points, frequent good users may very well be misclassified as malicious and removed.

Conversely, if the second method is used, the malicious party can exploit the system by providing a large number of points, each slightly divergent from the true values, so the average misfit will be much lower than most users and can safely avoid any detection.

The last method has the merit of not leaning towards either extreme, and more sophisticated versions can offer a tiered mismatch system that attempt to capture the pattern of the good users, and punch both single point offenders and mass attack offenders. Depending on the particular application, it is possible

to force the malicious party to disguise themselves by applying the same usage pattern as other users, thus limiting their capacities.

Multiple malicious users To remove multiple malicious users, one can either identify and remove the top several offenders rather than just topmost, or remove the top one, recalculated the solution using the data from remaining users, then again identify the biggest offender. The former is less computationally intensive, but the latter may yield better results because removing one malicious user's data may very well expose another malicious user's.

major challenge to the algorithm Apart from the obvious technical problem of not being able to find the optimal solution via a greedy approach, there is a more fundamental challenge to this algorithm that make it unsuitable for serious usage where adversary is a major source of concern. It motivates the third and most advanced solution.

4.4 Third Solution: random sampling and piecewise linear reconstruction

Motivation

In order to start the process of removing outliers, we must first generate a solution for comparison. Whatever method we use– whether it is using all data points from the ℓ^1 best fit as described earlier or some other way, as long as the initial solution contain malicious data points, then it is possible that organized malicious data overwhelm the good data points, so that the initial solution leans towards the malicious data points rather than the accurate one, and the process will simply repeatedly remove good users, not malicious ones. This highlights the fundamental issue that there is no signature separating the malicious data points from the good ones, and any calculation must start with a troublesome assumption.

For example, if one malicious user injects 100 points 10 units of time ahead of actual time, and 10 good users correctly report 5 points data points each, the biggest offenders may very well be the good users, and as more and more good users are removed from the data set, the malicious user appears to be more and more correct, and more and more good users are in jeopardy. In fact, in this extreme example there are more malicious points than the good ones and the two sets are clearly disjoint and

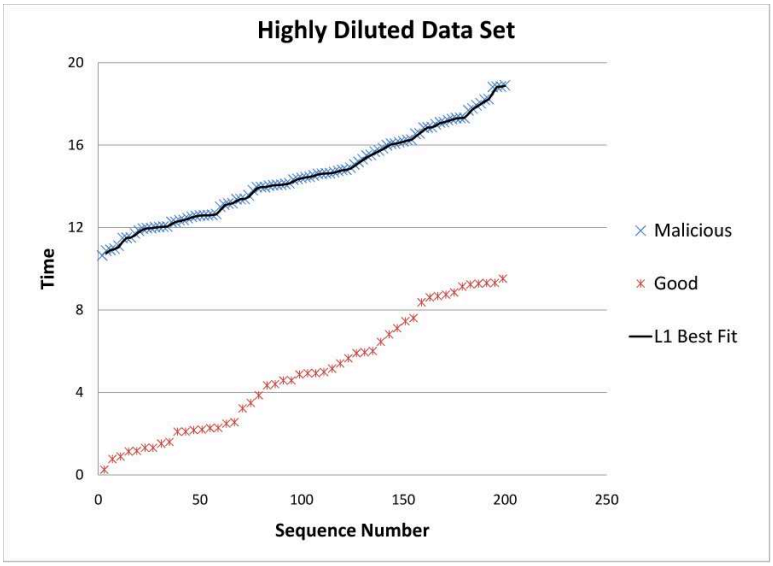


Figure 13: The blue X represent 100 points a malicious user reported to the server, and the red stars represent a total of 50 data points contributed by many good users. The ℓ^1 best fit curve actually perfectly go through every malicious point, demonstrating any method based on the entire data pool can result in highly misleading results.

impossible to consolidate, the ℓ^1 best fit will perfectly go through all the malicious data points as to achieve zero residual in those points, and ignore the good points altogether. See Fig. 13.

ℓ^2 best fit would lie somewhere between the two sets of data, closer to the malicious one because the number of points there is more numerous. Other variations on this solution may help in particular cases but they would still not be able effectively address and counter the malicious party's ability to spawn an overwhelming number of points.

Fundamentally speaking, the solution removes outliers, not malicious data points. While it is highly effective in reducing the total residual, there is no proof that decreasing residual is the same as reconstructing the correct time of presence for the clients.

Overview

Because of the possibility for a malicious user to swamp the data with falsified information, the deduced time must be based on the majority of users, not of data

points. Thus we redefine the problem to minimize the mismatch between the constructed data series and the users, not just data points. Based on this philosophy, we developed the following mechanism to derive actual time based on unreliable time stamps.

1. pre-screening filter
2. randomly sample a portion of the users, and use their data to construct a monotonic best fit time series. Repeat the process sufficiently many times.
3. construct a final solution by connecting the medium piece-wise linear segments of the best fit graphs.
4. validate the solution by examining how many users' data points are significantly at odds with the solution. Reconstruct the solution if the discrepancy is alarming.

Pre-screening

Pre-screening is deployed to quickly eliminate data points that are self-inconsistent and obviously erroneous. In particular, if a user reports multiple points such that some have a earlier time but later sequence number or if a data point reported is out of range, then the user must be experiencing some technical difficulties and is neither stable nor reliable. All the data from such users should be filtered out at this stage and not in anyway contribute to the construction of the solution.

Sampling

Random sampling is used to introduce more unpredictability for the malicious party hence limit what they can do. A small but significant percent of user, β, should be sampled for each iteration, and

These parameters will affect the efficiency and accuracy of the final solution, as discussed later. If a user is sampled, all the data from the user should be included and combined with data from other sampled user. After generating a monotonously non-decreasing solution from this data pool using least square fit, we can repeat the process to generate more sampled solutions. This process will create several sample curves, see Fig. 14 for an example. The process of deriving those best fit curves are similar to the ℓ^1 best fit solution described before, except ℓ^2 is used. This is because

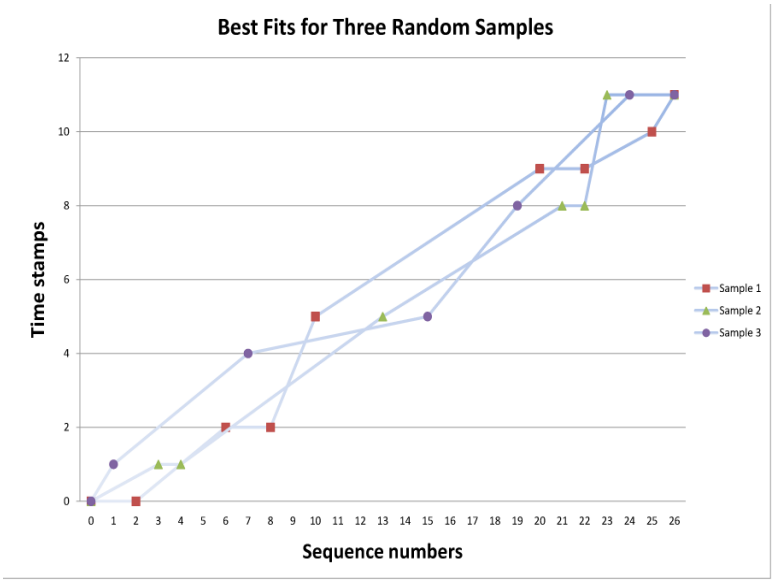


Figure 14: The three different curves are marked with different colors. The sequence number of each data point is marked on the x axis, while the adjusted time stamps are on the y axis.

ℓ^2 solutions tend to be stable and unique. Unlike before, where we tried to minimize the effect of outliers to protect the solution, exaggerated effects of the outliers is not a source of concern, because skewing the curve will reduce the probability for the incorrect segment to be selected in the final solution. This means that the malicious party cannot lower the accuracy of the final solution by altering any particular sample curves.

Final Construction

Each run of random sampling will generate one monotonically non-decreasing piecewise linear graph. At any point in time, there is a well-defined medium value among all the graphs. We can find a final solution based on these medium points. Even assuming the malicious users are organized into a systematic attack, and that close to half of the best fit curves contain malicious values, they would still have little impact on the construction of the final solution because it looks at only the medium, not the arithmetic mean.

For example, Fig. 14 shows the monotonic best fits of three randomly sampled curves. The final solution constructed using the three curves are shown in Fig. 15.

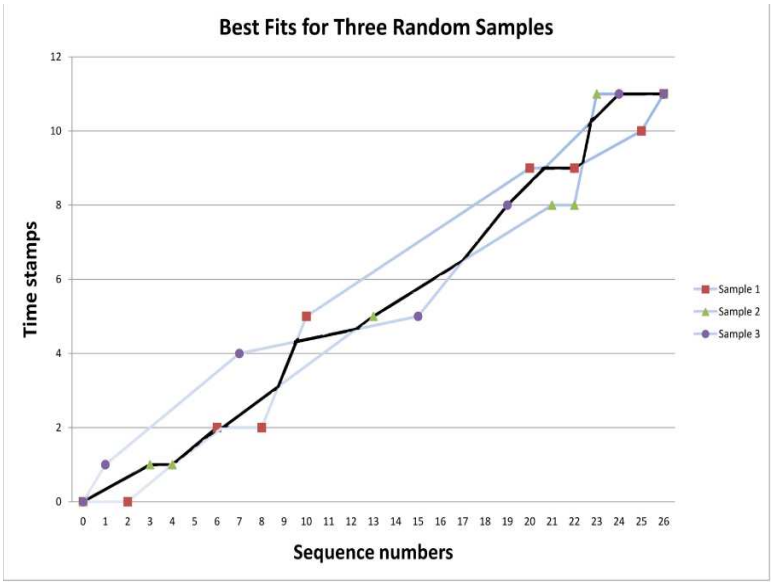


Figure 15: The same graph with the final solution based on the medium pieces of the three best fit curves.

Note the points on the graph are not necessarily what the data reported, but rather the adjusted values in order to satisfy the monotonicity.

If some of the best fit curves are affected by some malicious data, as in the case represented in Fig. 16, the affected segments of the curve would not be selected as part of the final solution because the segment would not be the medium of the curves. In fact, since it is unlikely for more than half of the best fit curves to be contaminated as shown in the discussion section of the protocol, the only way for any segment to be shown as part of the final solution is to be within the margin of error of good users' time points, hence the information provided is by definition correct.

In the particular example shown in Fig. 16, segments around the malicious data (sequence numbers 10-15) are not selected as part of the final solution (see Fig. 17), but data points 0, 1 and 2 on the same curve would be included.

Clearly other methods of consolidating the curves such as taking the arithmetic or geometric mean would result in a final solution seriously skewed by the introduced contaminated curve. In the design proposed in this thesis, It is irrelevant how many points the malicious party adds or how misleading they are, for the only way of increasing the chance of disturbing the system is to introduce more malicious users, which can be an onerous task as discussed in the adversary section. Even if a few

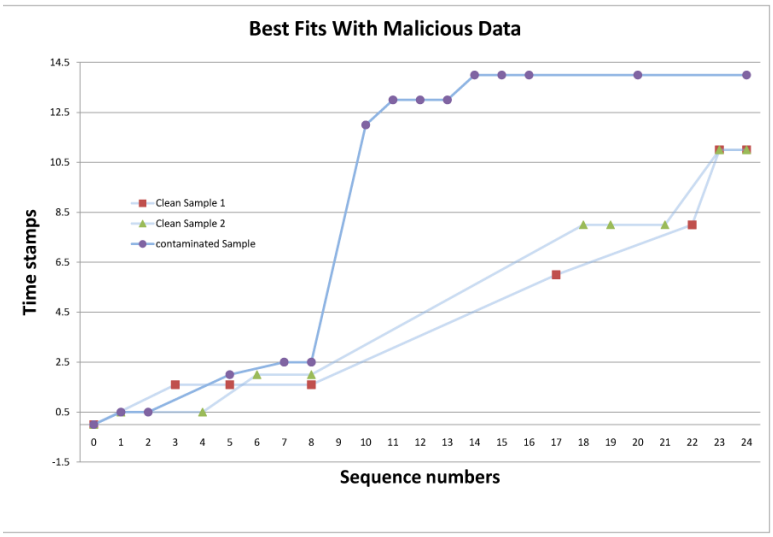


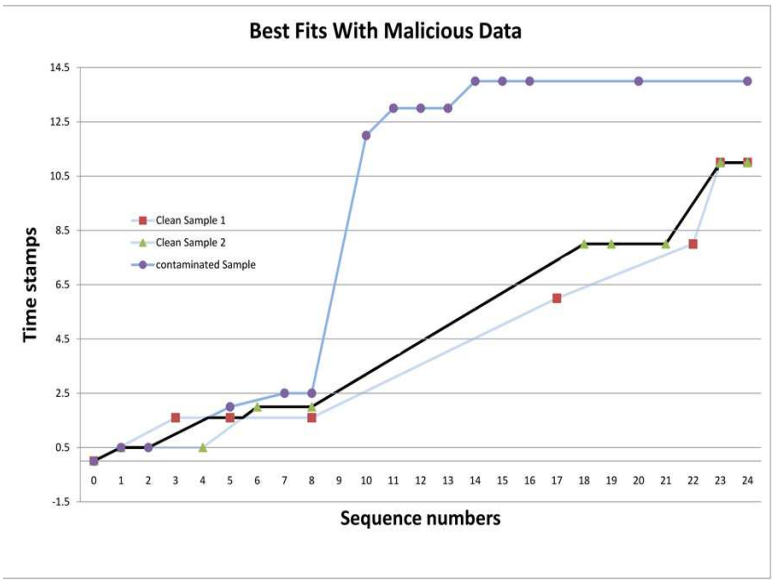
Figure 16: Data points 10 though 15 on the graph are submitted by a malicious user. The graph shows two samples that do not contain those points, and one sample that does. The malicious data points completely throw off the third curve.

malicious users' data account for a majority of all the data, their false information will only serve to skew a small percentage of the random samples, and the final solution constructed would not be affected by this type of attack.

Validation

It is unlikely but possible that the malicious users are repeatedly sampled by the randomized processes and have a significant over-representation hence heavily influence the final solution constructed following the steps outlined above. In particular, in the worst case, the malicious users can fully manipulate the final result if they falsify information intelligently and are lucky enough to contaminate more than half of the simple solutions used in the construction of the final solution.

As a countermeasure and in order to provide additional assurance of accuracy, we can use the data to validate the final solution. Under the reasonable assumption that the malicious users as a percentage of the total users cannot exceed some particular value α, we can easily detect if more than α percent users have at least one data point significantly different from what the solution would indicate. For example, in the unlikely event the final solution is similar to 13 by overweighting the malicious party's input, the validation process would dispose the solution on the group that it



[h]

Figure 17: This graph highlights, in black, the segments from Fig. 16 that would be included in the final solution. Note how the contaminated parts of the curve is not part of the final solution, and that does not discredit or disqualify other segments of the same curve, in fact from sequence number 0 to 4, the final solution is a trace of the accurate part of the contaminated curve.

fails to match the majority of users' data inputs.

This effectively guarantee that the system will always correctly record the time of presence for about $(100 - \alpha)\%$ of all users.

5 Future Research

It would be interesting to further explore and expand on the ideas presented in this chapter by

1. analyzing some real-life data, such as airport traffic, to apply the scheme and further verify the results;

2. looking at different sets of real-life data and observe common patterns that can help identify malicious users' typical behavior; and

3. Whether it is possible to predict future traffic flow and/or predict if particular individuals would appear in a certain location based on data from this protocol.

6 Summary

Through a fairly straightforward protocol and a series of progressively more complex solutions, we have shown in this chapter the feasibility of providing location proof service using RFID tags. The simplicity of the hardware proves to be a double-edged sword, because it makes it possible to provide very economically a service which generally require much more advanced hardware as well as more manpower to support, but at the same time, the simplicity of the hardware imposes serious challenges to the central processing unit which is responsible for consolidating and verifying data from local tags.

The most advanced solution presented in the chapter has many nice mathematical properties that suggest it should be relatively efficient and accurate, and allows for application- specific tweaks of parameters that can balance computational intensity and resolution of time stamps. The solution also defends against a wide array of malicious tactics. Outliers will not impact the final solution construction, and the solution verification scheme provides a level of confidence in the solution.

Part V

Conclusion

Designing efficient, secure and robust protocols can be a tremendous task, and the requirements vary depending on the level of hardware and the difficulty of the problem. This thesis attempts to isolate some of the real life applications where such open requirements result in additional challenges. For the vehicular sensor network, the focus was on the security of the sensor communication against common attacks such as eavesdropping. The resulting protocol, however, is not only limited to vehicular networks. The body sensor part does not contain extensive discussion of security, in part because the four-way handshake proposed in the vehicular network part of the thesis can be applied to this situation in order to provide a similar level of security. Similarly, the advanced processing proposed in the RFID chapter of the thesis can be used as a model to identify and correct possible security breaches.

The body sensor network part of the thesis has shown interesting potential in terms of conveying and inferring a large amount of information while uploading a trivial amount of data. As described in the prioritized delivery of information subsection, it is possible to further limit the number of communications if the central unit only queries some sensors on an as-needed basis.

The focus on efficiency in the first part differs from the focus on security in the vehicular sensor network, where experiments were carried out on this different application, the data from which both confirmed some assumptions and provided interesting, unexpected results. It was comforting to confirm that it takes only a fraction of the total amount of time available to complete the four-way handshake. Interestingly, the RSSI is not strongest when the car mote and roadside motes are right next to each other, but when the car mote is still one second away from it. One possible explanation of this is the Doppler Effect.

The last chapter presented here used RFID tags to stress the importance of central processing. It shows that through some intelligent combination of deterministic and random processing, it is possible to infer correct time series from a pool of data that contains a large portion of malicious inputs. This result is pleasantly surprising since, intuitively, the best we can do seems to be averaging all available information. It also reveals that many crucial parameters, such as the resolution of the time series, can

be left for administrators of specific systems to decide, making the protocol robust in nature.

Together, the lessons and results of the three applications form a general framework for designing protocols for sensor network systems. Future protocols for other applications can utilize many of the findings outlined in this paper. For example, the SIM data structure, four-way handshake, and the randomized multi-stage consolidation algorithm can all be applied to other applications. A mission-critical system can use several of the security and verification features presented in this paper to create a multi-layer failsafe protocol, leaving the malicious party with little room to intrude.

The first protocol also provides insights into generic methodologies of energy efficient designs, while the latter two applications approach the issue of security from different perspectives. The task of security is especially difficult compared with distributed server networks and other infrastructures made up of powerful devices because of the hardware weakness of the sensor networks. We believe that, on top of solving real life problems and providing meaningful solutions on their own, the three applications also serve to further our understanding of the general processes of protocol design for senor networks.

References

[1] M. Corporation, *Tmote Invent User Guide*, 2006.

[2] H. Junker, M. Stager, G. Troster, D. Blattler, and O. Salama, "Wireless networks in context aware wearable systems," in *Proceedings of the 1st European Workshop on Wireless Sensor Networks*, pp. 37–40, 2004.

[3] B. Lo and G. Yang, "Key technical challenges and current implementations of body sensor networks," *Body Sensor Networks*, 2005.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[5] R. Ganti, P. Jayachandran, T. Abdelzaher, and J. Stankovic, "Satire: a software architecture for smart attire," in *Proceedings of the 4th international conference on Mobile systems, applications and services*, pp. 110–123, New York, NY, USA, 2006.

[6] E. Farella, L. Benini, B. Riccò, and A. Acquaviva, "Moca: A low-power, low-cost motion capture system based on integrated accelerometers," *Advances in Multimedia*, 2007.

[7] A. Wood, J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru, "Context-aware wireless sensor networks for assisted living and residential monitoring," *IEEE Network*, vol. 22, no. 4, pp. 26–33, 2008.

[8] J. Perng, B. Fisher, S. Hollar, and K. Pister, "Acceleration sensing glove (ASG)," in *The Third International Symposium on Wearable Computers*, pp. 178–180, 1999.

[9] T. Stiefmeier, G. Ogris, H. Junker, P. Lukowicz, and G. Troster, "Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario," in *10th IEEE International Symposium on Wearable Computers*, pp. 97–104, 2006.

[10] P. Angkititrakul, M. Petracca, A. Sathyanarayana, and J. Hansen, "UTDrive: driver behavior and speech interactive systems for in-vehicle environments," in *2007 IEEE Intelligent Vehicles Symposium*, pp. 566–569, 2007.

[11] C. Tan, H. Wang, S. Zhong, and Q. Li, "Body sensor network security: an identity-based cryptography approach," in *Proceedings of the first ACM conference on Wireless network security*, pp. 148–153, ACM New York, NY, USA, 2008.

[12] P. Fergus, K. Kafiyat, M. Merabti, A. Taleb-bendiab, and A. El Rhalibi, "Remote physiotherapy treatments using wireless body sensor networks," in *IWCMC '09: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing*, (New York, NY, USA), pp. 1191–1197, 2009.

[13] E. Jovanov, A. Milenkovic, C. Otto, and P. C. de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation.," *Journal of neuroengineering and rehabilitation*, 2005.

[14] S. Shepard, *RFID: Radio Frequency Identification*. McGraw-Hill, 2005.

[15] H. Gao, S. Utecht, F. Xu, H. Wang, and Q. Li, "Experimental study on secure data collection in vehicular sensor networks," in *WASA '09: Proceedings of the 4th International Conference on Wireless Algorithms, Systems, and Applications*, (Berlin, Heidelberg), pp. 22–31, 2009.

[16] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddonand, and H. chi Wong, "Secret handshakes from pairing-based key agreements," in *IEEE Symposium on Security and Privacy*, pp. 180–196, 2003.

[17] R. Marchese, M. Diverio, F. Zucchi, and C. Lentino, "The role of sensory cues in the rehabilitation of parkinsonian patients: A comparison of two physical therapy protocols," *Movement Disorders*, pp. 879–883, 2000.

[18] R. R. S. Weis, S. Sarma and D. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," *1st Intern. Conference on Security in Pervasive Computing (SPC)*, 2003.

[19] M. Raya, P. Papadimitratos, and J. P. Hubaux, "Securing vehicular communications," in *IEEE Wireless Comm*, 2006.

[20] H. Chan and A. Perrig, "Pike: peer intermediaries for key establishment in sensor networks," in *INFOCOM*, pp. 524–535, 2005.

[21] W. Du, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, pp. 586–597, 2004.

[22] H. Vogt, "Exploring message authentication in sensor networks," in *Proc. of European Workshop on Security of Ad Hoc and Sensor Networks (ESAS), LNCS*, Springer-Verlag, 2004.

[23] S. Zhu, S. Setia, and S. Jajodia, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *IEEE Symposium on Security and Privacy*, pp. 259–271, 2004.

[24] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using WiFi," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, MobiCom '08, (New York, NY, USA), pp. 199–210, 2008.

[25] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular opportunistic communication under the microscope," in *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, (New York, NY, USA), pp. 206–219, 2007.

[26] H. Wang and Q. Li, "Distributed user access control in sensor networks," in *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 305–320, 2006.

[27] J. Landt, "The history of RFID," *Potentials, IEEE*, vol. 24, no. 4, pp. 8–11, 2005.

[28] B. Eckfeldt, "What does rfid do for the consumer?," (New York, NY, USA), pp. 77–79, September 2005.

[29] W. Luo and U. Hengartner, "Veriplace: a privacy-aware location proof architecture," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, (New York, NY, USA), pp. 23–32, 2010.

[30] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, HotMobile '09, (New York, NY, USA), pp. 3:1–3:6, ACM, 2009.

[31] L. D. J. X. Stojmenovi, I., "A scalable quorum-based location service in ad hoc and sensor networks," vol. 1, pp. 71–94.

[32] L. Bussard and W. Bagga, "Distance-bounding proof of knowledge to avoid real-time attacks," in *SEC*, pp. 223–238, 2005.

[33] J. T. Chiang, J. J. Haas, and Y.-C. Hu, "Secure and precise location verification using distance bounding and simultaneous multilateration," in *Proceedings of the second ACM conference on Wireless network security*, (New York, NY), pp. 181–192, 2009.

[34] G. P. Hancke and M. G. Kuhn, "An RFID distance bounding protocol," pp. 67–73, 2005.

[35] S. L. Garfinkel, A. Juels, and R. Pappu, "RFID privacy: an overview of problems and proposed solutions," *Security & Privacy Magazine, IEEE*, pp. 34–43, 2005.

[36] A. Juels, "Minimalist Cryptography for Low-Cost RFID Tags (Extended Abstract)," *Security in Communication Networks*, pp. 149–164, 2005.

[37] H. Lee and J. Kim, "Privacy threats and issues in mobile rfid," in *Proceedings of the First International Conference on Availability, Reliability and Security*, (Washington, DC, USA), pp. 510–514, 2006.