Audio-based Self-organizing Authentication for Pervasive Computing: a Cyber-Physical Approach

Su Jin Kim and Sandeep K. S. Gupta
IMPACT LAB (URL: http://impact.asu.edu)
Arizona State Universeity
Tempe, Arizona, USA
Email: {Su.Kim and Sandeep.Gupta}@asu.edu

***Abstract*—Pervasive computing is fast becoming a reality with rapid advance in computing and networking technologies. It has the characteristics of scalability, invisibility, and the absence of the fixed infrastructure. Thus, existing security solutions do not fit this new computing environment. In many pervasive applications, devices collaborate with others within a particular area (such as a room or a building) and thus such applications need authentication for security. In this paper, we propose a new authentication technique which allows devices to authenticate each other when they are within the same audio region. The main advantage of our scheme is that there is no need of the pre-shared secret, the pre-existing trust relationship and human involvement. This paper takes a cyber-physical approach and generates features from the environmental sound. The idea is that this feature is used to verify the legitimacy of devices. We present the audio-based self-organizing authentication scheme and experimental results showing the achievement of requirements. We also analyze the computation cost, energy cost and security.**

***Keywords*-security; authentication; pervasive computing; cyber-physical security solution;**

I. Introduction

Pervasive computing is a new computing model that integrates seamlessly with everyday objects and activities [1]. It consists of small and inexpensive computing devices embedded in any type of objects such as cars, home appliances, various consumer goods and human body. These embedded devices are connected via wireless networks or possibly wired networks. There are various examples of their use such as monitoring borders, health, the environment (e.g. pollutants, wildlife, etc.), and automated homes. The pervasive computing is rapidly becoming a reality with rapid advance in computing and networking technologies, but the security for such systems has not been adequately explored.

Authentication is a common security service which has been studied for a long time [2]. Existing authentication mechanisms for traditional computing environments are based on the user input (e.g., password or PIN), the trusted-third party (e.g., public key infrastructure or a key distribution centre) or the pre-shared secret [2]. In pervasive computing, however, these traditional solutions are not appropriate because of the following reasons. First, the number of devices will tremendously increase. Obviously, existing solutions that use pre-shared secret or user password will not scale to the huge computing environment [3]. Second, the ideal solution for the pervasive computing should be transparent to users [2]. Thus, the authentication should be an automatic and seamless process running in the background with minimal user interaction. Third, there is no fixed infrastructure, so the existing authentication scheme through the trusted third party is not feasible for this environment [4].

The goal of our work is to develop an authentication solution which does not require human involvement, the pre-shared secret, and the trusted third party. To solve this problem, we use the concept of *Cyber Physical Security Solutions (CPSS)*. CPSS is a new class of security solution which generates a secret from their physical environment [5]. In this paper, *we suggest to use environmental sound as the source of a secret. The secret derived from environmental sound is used to authenticate devices.*

Our contributions in this paper are as follows: First, we propose a *self-organizing* authentication scheme. It does not require any help of human, the trusted third party, and the pre-shared secret. Second, our mechanism provides *adaptability* to the characteristics of environments such as the computing capability, security requirement, and other factors. For example, the length of the feature can be reduced in a resource-constrained computing environment, but it can be increased with the high security requirement. Third, we investigate various factors of the proposed scheme and perform an experimental study. Particularly, we show the computation and energy cost of our scheme using an actual sensor device, Crossbow's TelosB motes [6].

The rest of the paper is structured as follows. In Section II, we explore the existing solutions of the authentication, especially proximity-based authentication and device pairing. Section III gives motivation and Section IV defines the problem and assumptions. In Section V, we explain the detail of our audio-based self-organizing authentication. In Section VI and VII, we describe the experimental results to evaluate our protocol. In Section VIII, we propose directions for future work and conclude.

II. Related Work

The Diffie-Hellman key exchange protocol [7] is a well-known key agreement protocol. It allows two entities to agree on a secret key over an insecure medium. However, the Diffie-Hellman key exchange protocol does not have an authentication process. Thus, it is vulnerable to a *man-in-the-middle attack* which monitors a communication between two parties and falsifies the exchanges to impersonate one of the parties. A number of enhancements [8], [9], [10] have been developed, but they used a public-key certificate or pre-shared password for authentication.

Recently several researchers have developed systems to authenticate and establish a key without pre-existing trust relationship such as public-key certificate and password. Stajano et al. have introduced the Resurrecting Duckling technique in [11]. When a new device wakes up, it is imprinted by the active master (like a duckling recognizes the mother). This imprinting must be done by the physical contact. However, the requirement of physical contact among all communicating devices will be too restrictive in our targeted applications.

McCune et al. [12] have developed the authentication mechanism using visual channel called as Seeing-is-Believing (SiB). SiB assumes a human user is capable of visually identifying the target device. In this approach, the sender first sends its key to the other via wireless channel. At the same time, it displays the hash of its key in the form of a barcode. On the receiver side, the device should have a photo camera. Using this photo camera, a user scans the bar code on the sender's display. Then, the receiver translates the scanned barcode into a binary key, and compares it with the key received over the wireless channel. One problem in SiB is that devices must have additional hardware (a display or a photo camera). Also, this lays burden upon a human user.

Some papers [13] [14] have proposed to use audio channel for authentication. In Loud-and-Clear [13], two devices exchange their keys over the wireless channel. The hash of the key is translated in syntactically correct (but usually nonsensical) English-like sentence. One device plays it and the other displays it. The human user compares them and verifies their keys. HAPADEP in [14] is similar to Loud-and-Clear, but it does not need a display. After key exchange, both devices play the key as sound. A user verifies the key by listening to playing sound. Clearly, these schemes are not suitable to our targeted applications because of the reliance of human assistant. Another drawback is that translating and encoding (to sound) are too expensive for the resource-limited mobile devices.

Amigo system by Varshavsky in [15] has used dynamic characteristics of the radio environment. Two devices first listen to their radio environment and derive a shared secret from it. If these devices are in close proximity, they will share the same secret. The advantage of this technique is that human involvement and additional hardware are not required. However, the performance of this technique highly depends on the types of WiFi cards and network conditions, such as WiFi usage. Some environments may have no radio signals or not vary over the time so that attackers can easily learn about the radio environments.

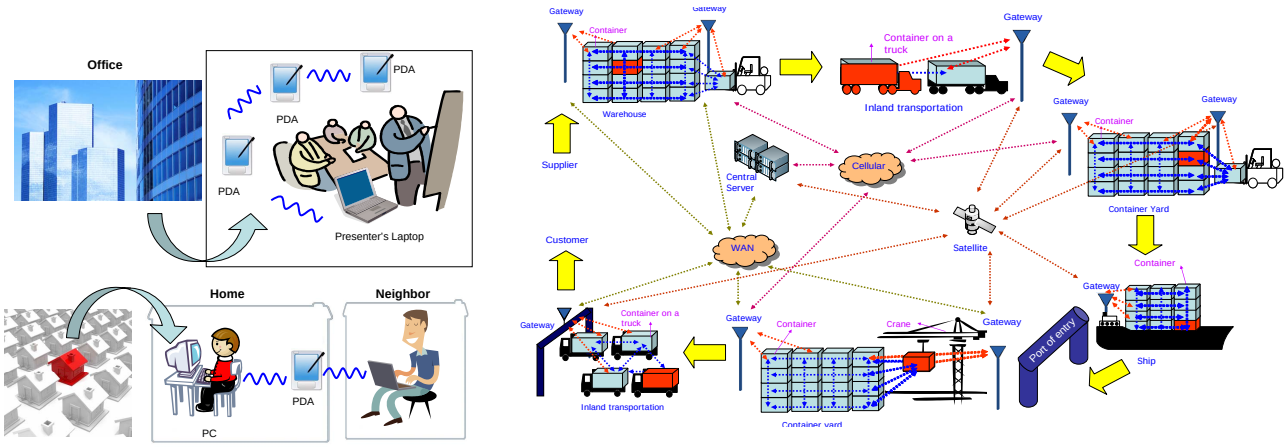
III. Motivation

In many pervasive applications, users/devices commonly form social ephemeral groups to collaborate with others [16]. The group membership is changed based on some social context, such as user's location, identity and interest. Currently, the location is the most widely used type of context for various applications. There are many examples of location-based services such as location-based billing, location-baed authentication, resource tracking, and Location-based mobile advertising.

This section gives two example scenarios of location-aware applications: the smart space and the intelligent container.

The smart space is one of most popular applications in pervasive computing. It includes a wide range of applications from home automation, health care, and education applications to game/entertainment applications. In a smart space, a user's devices can join a local network and collaborate with its current neighbors. Figure 1(a) shows one example scenario of smart space applications. Suppose a person has a meeting with co-workers in his office. The participants' mobile devices (e.g. PDAs, laptops, and cell phones) can automatically form a group and exchange documents and their contact information. This group membership must be dynamically changed based on the user's location. For example, after work, his mobile devices should join his home network. Through this network, it can transfer the files and other data to his desktop at home.

Another example is the intelligent container system [17] [18] that we have developed to enhance the homeland security and global supply chain management. The goal of this project is to develop a real-time monitoring system for cargo containers. Unlike the existing systems focusing on the security of each individual container in isolation, we suggested the creation of a dynamic mesh network among neighbor containers. These mesh networked containers can take advantage of the fact that containers are in close physical proximity while in transit or stacked in container yards or in the port shown as Figure 1(b). As an instrumented container moves from one location to the next, its participation in the current mesh network will occur automatically through the self-configuring nature of the network. This mesh networked container provides end-to-end visibility throughout its lifecycle. We believe that the security of containers will be enhanced by this interaction between neighboring networked containers. For example,



(a) Smart Space

(b) Intelligent container

Figure 1. Motivation Scenarios

the container network can routinely ensure the segregation of incompatible materials through automatic sensing and interchange of information via the mesh.

Considering these example scenarios, most devices are connected over wireless networks. Wireless networks are more vulnerable than wired networks because of the broadcast nature of wireless communications. Thus, we need to protect the sensitive data transmitted. For example, in the case of smart space, the data transmitted between devices can be confidential documents or personal contact information. Suppose a neighbor has a powerful antenna and sits within the communication range. Then, he can easily eavesdrop and learn the secret data. More serious forms of attacks are active attacks that attempt to modify the data or access resources. For example, a thief can turn off a security alarm in the house to enter. Obviously, a legitimate user does not want to allow other people to hear his communication, and access his resources. Therefore, authentication is required before communication, collaboration and access grant.

In the above scenarios, a device should be within a particular area (e.g., a room, a building, a ship, or a port) in order to join a local network and collaborate with other devices. In order words, location information is used to authenticate a device. In this paper, *we focus on the authentication problem which is the process to prove one's identity and authenticity by detecting its current region.*

IV. Problem Definition and Assumptions

In this paper, we only consider applications which require authentication based on user's current region. As we discussed in Section III, devices within a particular region form a group and collaborate with each other. Therefore, devices must be authenticated by each other only when they are within the same region.

We define the problem of *self-organizing authentication: a device automatically authenticates a requester when they are in the same region*. The region can be a room, a house, a building, a ship, a yard or other physical areas. We assume that the requester claims that they are present in the particular region rather than a particular point of location. This region must have some sort of physical control to restrict people into this region. In other words, only authorized people are allowed to enter the region.

Both devices are mobile with wireless communication capability. When devices are present in the same region, we call them *co-located*. If the requester sends a message, the participant must be able to receive it and then both devices must behave according to the protocol. The requester will be authenticated if they are co-located and the requester behaves according to the protocol. Otherwise, the claim will be rejected.

Authentication must be an automatic process. There must be no human involvement so that a user can concentrate on his/her primary work. Futhermore, devices do not need any pre-shared information or pre-established trust relationship.

According to the above assumption, an adversary should not actually have any presence inside the restricted region. However, there can be malicious nodes. In this work, we consider an active adversary. The attacker can capture, record, intercept, replay or insert any message in the wireless communication medium using a powerful antenna. Therefore, the attacker can record the previous communications and replay valid packets to pair with other legitimate nodes (which are located within the region) as *replay attack*. In addition, there could be the attacker *guessing* valid packets. The attacker can also try to *impersonate* a legitimate node by intercepting all messages and injecting a new message. It is called as *man-in-the-middle attack*.

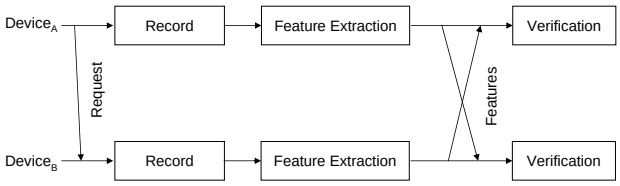


Figure 2. Proposed authentication protocol

V. Proposed Audio-based Self-organizing Authentication

In this paper, we suggest to take the *cyber-physical approach* in order to solve the self-organizing authentication problem. Especially, we propose to use environmental sounds to take the advantages of the following facts.

- Devices within the particular region hear similar environmental sound.
- Microphone is very cheap and many contemporary devices are equipped with a microphone.
- Environmental sound is produced by real random events in any physical state in any frequency range.

Figure 2 shows how our authentication process works. Suppose $Device_A$ wants to initialize communication with $Device_B$. First, $Device_A$ sends the request. Then, both $Device_A$ and $Device_B$ record sound and extract features from recorded sound. After feature extraction, $Device_A$ and $Device_B$ exchange the features and compare them. If the features are similar, authentication is done successfully. Otherwise, authentication fails.

A. Recording Phase

To achieve the accurate measurement, two devices should be synchronized well. There are several time synchronization algorithms available in the literature. In [19], high-precision clock such as 1 μsec can be achieved using small wireless sensors. We assume the devices use this synchronization technique to support $\approx$ 1 μsec high-precision clock.

The sampling rate is one of the critical factors in the recording phase. The sampling rate is the number of samples taken per second. By the Nyquist theorem, we must have at least two samples in each cycle: one measuring the positive part of the wave and one measuring the negative part. More than two samples per cycle will increase the amplitude accuracy, but less than two samples will cause the frequency of the wave to be completely missed. Audio data consists of time series which are usually quite large. Storing and computing such large data are costly, especially for resource-limited embedded devices.

The sampling duration is another important factor in this phase. Apparently, longer sampling durations can improve the performance of this protocol. In pervasive applications, however, authentication should not interrupt users as well as other services.

Because *there is the trade-off between accuracy and distraction*, the experimental study is necessary to determine appropriate sampling rate and duration. In this work, we tested various sampling rates and durations. The experimental results are in Section VII.

B. Feature Extraction Phase

The feature extraction phase analyzes the incoming waveform and extracts certain features from it. The feature should meet the following requirements:

- *Similarity*: when $Device_A$ and $Device_B$ are co-located, they should share the common feature(s).
- *Distinctiveness*: When $Device_A$ and $Device_B$ are not co-located, the feature(s) derived by $Device_A$ should differ from $Device_B$.
- *Randomness*: the feature(s) should be random so that no one can predict it.
- *Time variance*: the feature(s) should vary over time to avoid the reusability.

We use the basic feature representation using *Fast Fourier Transform (FFT)* analysis. The Discrete Fourier Transform (DFT) function implements the transform of the feature vectors of length N by:

$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)} \quad (1)$$

where $\omega_N = e^{(-2\pi i)/N}$ is an Nth root of unity. It is widely used in signal processing to analyze the frequencies contained in a sampled signal. DFT can be computed efficiently in practice using a FFT algorithm.

After applying FFT, N-point FFT (length N) can produce $\frac{N}{2}$ unique features because there is an even symmetry around the center point. In case of Matlab, each point contains a 4-bytes value (double data type), so a final feature set will be $\frac{N}{2} \times 4$ bytes.

C. Feature Exchange Phase

In the feature exchange phase, the problem is how to exchange features securely. If the feature has high randomness and time-variance, it can be directly sent. However, it is very difficult to get highly random and time-variant features over the short time period.

Existing key establishment solutions can be extended with our authentication mechanism to solve this problem. For example, suppose two devices exchange a key using the Diffie-Hellman protocol. A device encrypts the features with its id and the nonce sent by the pair using the key. It then sends the encrypted message to the pair. By the proof in [7], the Diffie-Hellman protocol supports secure key exchange. Since the message can not be decrypted without the key, an eavesdropper can not get the feature. We will analyze security of our protocol in Section VII-E.

D. Verification Phase

After exchanging extracted features, two devices will verify co-location of two devices based on the similarity of features. In this paper, we use *cross-correlation* which is a measure of the similarity of two signals. The cross-correlation coefficients, $crosscorr_k$ can be defined as a function of the time-lag k [20].

$$crosscorr_k = \frac{\sum_{t=1}^{N-k}(x_t - \overline{x})(y_{t+k} - \overline{y})}{\sqrt{\sum_{t=1}^{N-1}(x_t - \overline{x})^2}\sqrt{\sum_{t=1}^{N-1}(y_t - \overline{y})^2}} \quad (2)$$

where x_t and y_t are signals obtained at time t, $1 \leq t \leq N$, and $\overline{x}$ and $\overline{y}$ are the mean of x and y values respectively. $crosscorr_k$ will lie in the range [-1, 1], with 1 indicating perfect correlation and -1 indicating perfect anti-correlation (the inverse of one of the series). When the cross-correlation coefficients become close to 0, they are not co-related. For this work, we do not consider the negative correlation as co-location.

If $crosscorr \geq$ a threshold, authentication succeeds. The threshold depends on the types of applications and thus it should be determined by experiments carefully.

VI. Experimental Setup

To evaluate our protocol, we collected data in four environments, 1) cafe, 2) cassroom, 3) house and 4) office. Two recorders ($Device_A$ and $Device_B$) were located within 10 inches and another ($Device_C$) was located outside like Figure 3. Recorded sounds were sampled at 256 Hz and 8 kHz. For each type of environments, we recorded sounds every 1 minute for about an hour (40 - 60 trials). Experimentally, we could not see any remarkable changes on the result with increment of sampling durations from 10 seconds to 1 minute. We used the sampling duration of 10 seconds because of the reasonable result and delay. We used the fft() function for feature extraction and the corrcoef() function for verification in Matlab.

Figure 4 shows the results of 2^{17}-point FFT recorded at 8 kHz in the house. In Figure 4(a) and 4(b), we can see there are similar patterns on the results of 2^{17}-point FFT because $Device_A$ and $Device_B$ are co-located (within the same region). However, the patterns in Figure 4(c) are different from them.

VII. Evaluation

In this section, we present the experimental results, evaluate our scheme, and analyze the security of our scheme.

A. Accuracy

Similarity and distinctiveness are related to both the performance and security aspects. To evaluate the satisfaction of these two requirements, we use *false positive* and *false negative rates*.

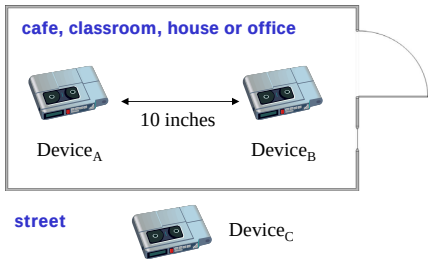


Figure 3. Experimental Setup

Table I
False Negative Rate (FNR) and False Positive Rate (FPR) with different lengths of FFT and Sampling Rates (SR)

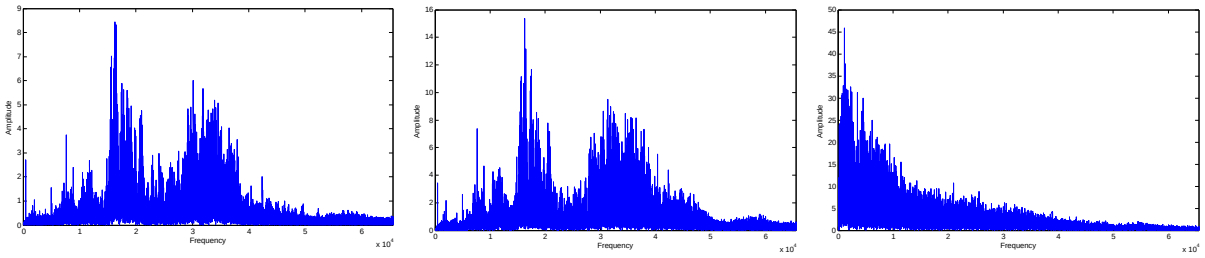
	Cafe	Classroom	House	Office
2^8-point FFT SR=256Hz	FNR=0 FPR=0.025	FNR=0 FPR=0	FNR=0 FPR=0.1	FNR=0 FPR=0
2^{13}-point FFT SR=8kHz	FNR=0 FPR=0	FNR=0 FPR=0	FNR=0 FPR=0.05	FNR=0 FPR=0
2^{17}-point FFT SR=8kHz	FNR=0 FPR=0	FNR=0 FPR=0	FNR=0 FPR=0	FNR=0 FPR=0

- *False negative*: the error of failing to reject authentication when it is in fact false.
- *False positive*: the error of rejecting authentication when it is actually true.

In a perfect authentication system, both rates should be zero. The false negative rate is more critical because non-zero false negative rates mean that systems are vulnerable. For the case of false positives, it can be solved by re-trial.

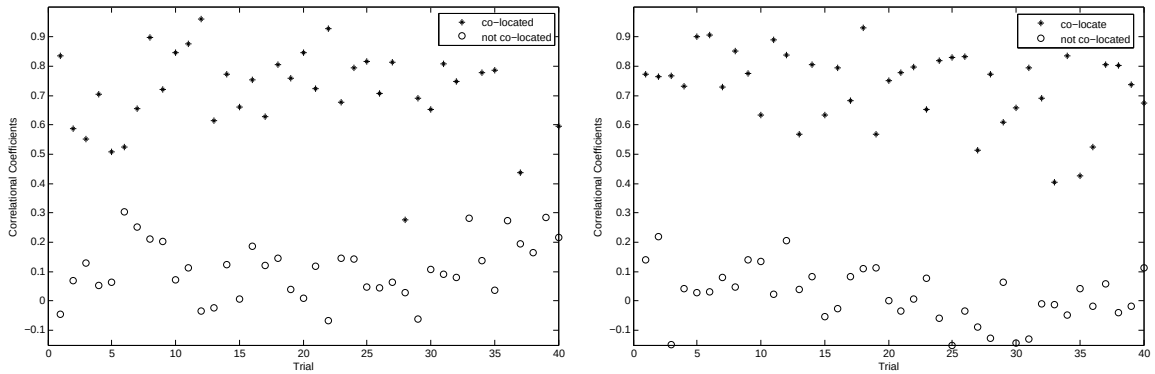
We performed 2^8-point FFT for 256 Hz and the 2^{13}-point and 2^{17}-point FFT for 8 kHz in order to find the effects of the sampling rate and the length of the FFT function. By the experimental results, we carefully chose threshold of 0.35 for 256-point FFT and 0.65 for 2^{13}-point and 2^{17}-point FFT. Table I shows the results with different lengths of FFT and sampling rates. Obviously, larger sizes of FFT with higher sampling rates provide more accurate results. With 2^{17}-point FFT, we achieved the perfect results for all test cases. However, FFT is very expensive to resource-limited embedded devices. The small number of re-trial with short delay might be acceptable in our targeted applications. Because the false positive rate of 0.1 could be acceptable for our cases, we used the 2^8-point FFT feature extraction with 256 Hz sampling rate for the rest of experiments. As we mentioned in Section I, the length of features and the sampling rate can be adjusted based on the characteristics of applications or devices (e.g., the level of security requirement and the available resources).

Figure 5 shows the correlational coefficients of 2^8-point FFT feature extraction with 256 Hz sampling rate. The stars and circles indicate the correlational coefficients between $Device_A$ and $Device_B$ (co-located) and between $Device_A$



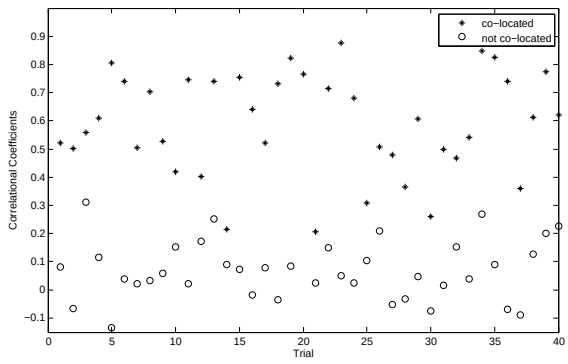
(a) $Device_A$ (b) $Device_B$ (c) $Device_C$

Figure 4. FFT results recorded in the house

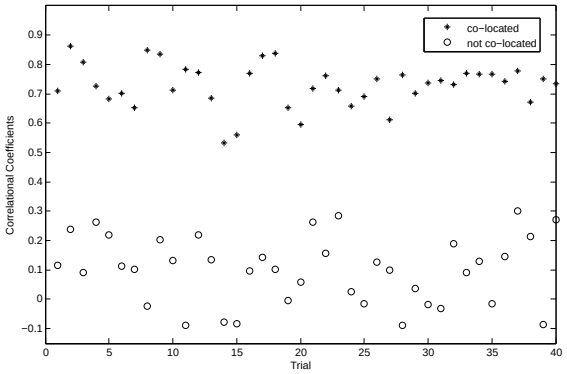


(a) $Cafe$

(b) $Classroom$



(c) $House$



(d) $Office$

Figure 5. Correlational Coefficients with 2^8-point FFT

and $Device_C$ (not co-located), respectively.

B. Computation and Energy Cost

To estimate the computational cost of 256-point FFT, we implemented it on Crossbow's TelosB motes [6] with an 8 MHz processor. More details can be found in [5]. Table II shows the results of 256-point FFT computation. Our code size of the 256-point FFT implementation is 15372 (about 15K) Bytes and it takes 440mSec. Energy is one of important factors in embedded system design. As our experimental results, 256-point FFT computation consumes 1mA with radio-off and 19.56mA with radio-on. A typical 1.5V AA battery has a capacity of about 2600 mAh [5]. With two AA batteries, a TelosB mote can compute the 256-point FFT function about 313 times with radio-on and about 6136 times with radio-off.

C. Randomness

To make sure the features are unpredictable by an attacker, we performed randomness tests using one of popular randomness testing program, "ENT" [21]. ENT performs a variety of tests on the stream of bytes as an input file and produces output as follows [21]:

- *Entropy* is a measure of the uncertainty associated with

Table II
CURRENT DRAW AND TIME RESULTS OF TELOSB'S 256-POINT FFT COMPUTATION

	Time	Current Draw	Total Energy Cost
Radio-off	440mSec	1mA	1mA×440msec×3V =1.32mJ
Radio-on	440mSec	19.56mA	19.56mA×440msec×3V ==25.81mJ

Table III
RESULTS OF THE RANDOMNESS TESTS FOR AN INDIVIDUAL FEATURE

	Cafe	Classroom	House	Office
Entropy	7.57	7.58	7.52	7.54
Chi-square Test (the number of success)	0	0	0	0
Arithmetic Mean	126.28	126.20	128.30	127.18
Monte Carlo Value for Pi	3.12	3.12	3.12	3.14
Serial Correlation Coefficient	0.75	0.60	0.89	0.39

a variable, expressed as a number of bits per character. With fully random inputs, the entropy should be 8 (bits per byte).

- *Chi-square Test* is a statistical test commonly used for the randomness test. The ENT program interprets the percentage as the degree to which the sequence tested is suspected of being non-random. If the percentage is greater than 99% or less than 1%, the sequence is almost certainly "not random". If the percentage is between 99% and 95% or between 1% and 5%, the sequence is "suspect". Percentages between 90% and 95% and 5% and 10% indicate the sequence is "almost suspect". For other percentages, we consider it passes the test successfully.
- *Arithmetic Mean* is the result of summing the all bytes in the file and dividing it by the file length. If the data is close to random, this should be around 127.5.
- *Monte Carlo Value for Pi* is also generally used as randomness test. In this program, each successive sequence of six bytes is used as 24 bit X and Y coordinates within a square. If (X, Y) point is inside a circle inscribed within the square, it is considered as "hit". With large streams, the percentage of hits called Pi will approach the correct value of π if the sequence is close to random.
- *Serial Correlation Coefficient* measures the correlation of successive bytes in the file. For random sequences, this value will be close to zero.

We performed the ENT program with an individual feature and then calculated the average of values for each environment. The results are shown in Table III. In all types of environment, features passed the most of tests, such as *Entropy, Arithmetic Mean*, and *Monte Carlo Value for pi* tests. However, all of them failed in *Chi-square* and *Serial Correlation Coefficient* tests.

Table IV
RESULTS OF THE "ENT" PROGRAM FOR THE SEQUENCE OF FEATURES OVER TIME

	Cafe	Classroom	House	Office
Entropy	7.57	7.58	7.52	7.54
Chi-square Test (the number of success)	0	0	0	0
Arithmetic Mean	126.28	125.75	128.30	127.10
Monte Carlo Value for Pi	3.12	3.13	3.12	3.14
Serial Correlation Coefficient	0.70	0.43	0.81	1.53

D. Time-variance

To test the time-variance of features, we used the ENT program again. The program ran with the sequence of all features collected over time (approximately one hour). The results are shown in Table IV. Similar to the results in Section VII-C, the sequence of features succeeded in *Entropy, Arithmetic Mean*, and *Monte Carlo Value for pi* tests, but failed in *Chi-square* and *Serial Correlation Coefficient* tests.

E. Security Analysis

This paper aims to secure a pervasive system against *man-in-the-middle attacks, replay attacks* and *guessing attacks*. In this section we discuss how our authentication technique prevents these attacks.

- *Replay attack*: If features do not have temporal variations, the attacker can capture the valid features from the previous communications and simply reuse them to get authenticated. Experimentally, we found that the features might not vary over the short time period (approximately, an hour). However, we proposed extended Diffie-Hellman key agreement protocol in Section V. Assume a device carefully selects a random number as a nonce. Then, the attacker can not copy and reuse a valid transmission.
- *Man-in-the-middle attack*: As the experimental results in Section VII-A, a device which is located outside of the region can not sense a valid feature. Therefore, it can not generate messages with the valid feature in order to impersonate a legitimate user.
- *Guessing attack*: means guessing valid features in order to impersonate a co-located device. In our experiments, a feature with 256-point FFT is 128 $\times$ 4 bytes. There are 2^{4096} possibilities to guess a valid feature. With the limited reply time and number of trial, this type of attacking can be protected.

VIII. FUTURE WORK AND CONCLUSION

In this paper, we proposed an audio-based self-organizing authentication protocol for embedded devices in pervasive computing. We described the recording, feature extraction,

feature exchange, and verification phases in detail and discussed the challenging problems in each phase. The FFT based feature extraction was tested with simple data set to verify our scheme. The experimental results showed that our scheme is able to distinguish co-located devices from non co-located devices perfectly with 2^{17}-point FFT feature extraction with 8kHz sampling rate. 2^{12}-point FFT with 8kHz sampling rate and 2^8-point FFT with 256Hz sampling rate produced the average false positive rates of 0.0125 and 0.03125 respectively. However, it can be solved by re-trial. Both cases provided zero false negative rate, so there is no chance to get authenticated by non co-located devices.

We implemented 256-point FFT using Crossbow's TelosB mote and estimated the computation and energy cost. We discussed a trade-off between accuracy and complexity and also analyzed the security of our authentication protocol against possible attacks.

This paper used the basic frequency-domain feature extraction technique. Experimental evaluation showed that the proposed feature extraction might not be random and time-variant. This is a challenging problem because an attacker might record the environmental sound and learn about the acoustic features for a specific region. To overcome this problem, we are working on different types of features, especially time-domain features in future work. Another potential vulnerability faced by the protocol is when a legitimate node executes the Diffie-Hellman protocol with an adversary before exchanging the acoustic features. This can potentially allow the adversary to reuse features from a previous execution of the protocol and authenticate successfully. To overcome this challenge, we are investigating ways to improve the time-variance of acoustic features, so that the knowledge of features at a given time cannot be reused at another. Finally, we are also working on experimenting with large data set as well as other types of randomness tests.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation Grant CNS-0831544 and Intel Corp. We would like to thank to Ayan Banerjee, Georgios Varsamopoulos and Krishna-Kumar Venkatasubramanian from IMPACT Lab. at Arizona State University for their valuable inputs.

REFERENCES

[1] F. Adelstein, S.K.S. Gupta, G.G. Richard III, and L. Schwiebert, "Fundamentals of Mobile and Pervasive Computing," McGraw-Hill, 2004.

[2] R. Campbell, J. Al-muhtadi, P. Naldurg, G. Sampemane, and M.D. Mickunas,"Towards security and privacy for pervasive computing," in Proc. Intl. Symposium on Software Security, Tokyo, Nov. 2002.

[3] D. Hutter, Security in Pervasive Computing, LNCS, Berlin, New York, Springer, 2003.

[4] A. A. Pirzada, and C. McDonald, "Secure pervasive computing without a trusted third party," in Proc. IEEE/ACS Intl. Conf. on Pervasive Services, 2004.

[5] K. Venkatasubramanian, A. Banerjee, S. K. S. Gupta, "Green and Sustainable Cyber Physical Security Solutions for Body Area Networks," In Proc. 6th Workshop on Body Sensor Networks (BSN'09), Berkeley, CA, June 2009.

[6] TelosB Datasheet from Crossbow, [Online]. Available: http://www.xbow.com/Products/productdetails.aspx?sid=252

[7] W. Diffie and M. E. Hellman, "New directions in cryptography," IEEE Trans. on Information Theory, pp. 644-654, 1976.

[8] W. Diffie, P.C. van Oorschot, and M.J. Wiener, "Authentication and authenticated key exchanges," in Designs, Codes and Cryptography 2, pp. 107-125, 1997.

[9] S. M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," in Proc. IEEE Symposium on Research in Security and Privacy, Oakland, May 1992.

[10] V. Boyko, P. MacKenzie, and S. Patel, "Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman," in Advances in Cryptology: Eurocrypt 2000, LNCS vol. 1807, Springer-Verlag, pp. 156-171, 2000.

[11] F. Stajano, and R. Anderson, "The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks." in Proc. 7th Security Protocols Workshop, pp. 172-182, 2000.

[12] J. M. McCune, A. Perrig, and M. K. Reiter, "Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentication." in Proc. IEEE Symposium on Security and Privacy, pp. 110-124, 2005.

[13] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and Clear: Human-verifiable authentication based on audio," in Proc. 26th IEEE Intl. Conf. on Distributed Computing Systems, pp. 10-17, 2006.

[14] C. Soriente, G. Tsudik and E. Uzun, "HAPADEP: human-assisted pure audio device paring," in Proc. 11th Information Security Conf., 2008.

[15] A. Varshavsky, A. Scannell, A. LaMarca, and E. Lara, "Proximity-Based Authentication of Mobile Devices," in Proc. 9th Intl. Conf. on Ubiquitous Computing, pp. 253-270, 2007.

[16] B. Wang, J. Bodily, and S. K.S. Gupta, "Supporting persistent social groups in ubiquitous computing environments using context-aware ephemeral group service," in Proc. 2nd IEEE Annual Conf. on Pervasive Computing and Communications, 2004.

[17] S. J. Kim, G. Deng, S. K.S. Gupta and M. Murphy-Hoye, "Intelligent Networked Containers for Enhancing Global Supply Chain Security and Enabling New Commercial Value," in Proc. 3rd Intl. Conf. on Communication System Software and Middleware, 2008.

[18] S. J. Kim, G. Deng, S. K.S. Gupta and M. Murphy-Hoye, "Enhancing Cargo Container Security during Transportation: A Mesh Networking Based Approach," in Proc. 2008 IEEE Intl. Conf. on Technologies for Homeland Security, 2008.

[19] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in Proc. 5th Symposium on Operating Systems Design and Implementation, Boston, USA, Dec. 2002.

[20] Cross Correlation. [Online]. Available: http://local.wasp.uwa.edu.au/ pbourke/miscellaneous/correlate/

[21] ENT: A Pseudorandom Number Sequence Test Program. [Online]. Available: http://www.fourmilab.ch/random/