Adaptive Wireless Channel Probing for Shared Key Generation

Yunchuan Wei* †, Kai Zeng† and Prasant Mohapatra†
*School of Automation
Beijing Institute of Technology, Beijing, China 100081
Email: weiyunchuan@bit.edu.cn
†Department of Computer Science
University of California, Davis, CA 95616
Email: {ycwei,kaizeng,pmohapatra}@ucdavis.edu

***Abstract*—Generating a shared secret key between two parties from the wireless channel is of increasing interest. The procedure for obtaining information from wireless channel is called channel probing. Previous works used a constant channel probing rate to generate a key, but they neither consider the tradeoff between the key generation rate (KGR) and channel resource consumption, nor adjust the probing rate according to different scenarios. In order to satisfy users' requirement for KGR and to use the wireless channel efficiently, we first build a mathematical model of channel probing and derive the relationship between KGR and probing rate. Second, we introduce an adaptive channel probing system based on Lempel-Ziv complexity (LZ76) and Proportional-Integral-Derivative (PID) controller. Our scheme uses LZ76 to estimate the entropy rate of the channel statistics (e.g., the Received Signal Strength (RSS)) and the PID controller to control the channel probing rate. Our experiments show that this system is able to dynamically adjust its probing rate to achieve a desired KGR under different moving speeds, different mobile types, different sites and different desired KGRs. Our results also show that the standard deviation of the LZ76 calculator is less than 0.15 bits/s. The PID controller is able to stabilize the key generation rate at a desired value with mean error of less than 0.3 bits/s.**

I. Introduction

Generating a shared secret key between two parties via public communication is a challenging problem in symmetric key cryptography systems. Diffie-Hellman (D-H) key exchange protocol is widely used for this purpose. However, it works under the assumption of the hardness of the discrete logarithm problem, which has been proven breakable in polynomial time using quantum computers [1]. Although realistic quantum computers may not become reality in years, it is desirable to search for other key agreement mechanisms which do not depend on computational power. Furthermore, in practical implementations, D-H key exchange protocol may not produce a truly random key due to the use of pseudorandom generators. With the spur of wireless communications, there is an increasing interest in generating a shared key from the wireless channel between two parties [2]–[5]. Two wireless entities exploit reciprocal and location-specific properties of a wireless fading channel, and obtain highly correlated channel states and produce identical symmetrical shared secret keys. A third party (that is more than half a wavelength away from the legitimate users) can eavesdrop but would not be able to generate the same key [6]. Therefore, unlike the D-H key exchange protocol, generating keys from the wireless channel is information theoretic secure, i.e. no matter how much computing resources the attacker has, the attacker cannot break the key.

In recent implementations and experiments, the received signal strength (RSS) is widely used as the parameter from wireless channel to generate the shared secret key. The RSS can be easily obtained from current wireless device drivers, so it makes key generation using off-the-shelf devices feasible. We call this process *channel probing*.

As far as we know, most related works probe the channel at a preset and constant rate without any consideration for channel variation. Usually, they prefer to set a high probing rate to generate a secret key as soon as possible. However, even though a user could get many frames and a long RSS sequence, a large part of it will have consecutive duplicate RSS values and will be discarded. Thus, these extra packets waste wireless channel resources and increase the cost of the key generation. On the other hand, it will take an intolerably long time to generate a key when probing at a very low rate.

As users always have requirements about how much time they can afford to generate a N-bit long key, we could control the probing rate to satisfy the *key generation rate* (KGR) constraint. In other words, the system does not have to probe too fast to get a high KGR; only fast enough to avoid using the channel inefficiently.

The KGR is partly determined by the quantity of information from the RSS sequence. The quantity of information is commonly measured by *entropy*, proposed by Shannon in 1948 [7]. Furthermore, the *entropy rate* or source information rate of a stochastic process is, informally, the time density of the average information in a stochastic process [8]. Thus, the KGR is fundamentally determined by the entropy rate and probing rate.

In this work, we build a mathematical model of the channel probing system and derive the relationship between key generation rate and probing rate. When the probing rate increases, the KGR increases but efficiency decreases.

In experimental situations, the computation of entropy rates requires a statistical estimator that is unbiased and converging fast enough to be accurate on a finite data sample. Unfortunately, since the classical definition of entropy rate is based on an asymptotic limit, it does not easily lead to an accurate estimator in the case of a finite-size time series [9]. The concept of Lempel-Ziv complexity (LZ76) [10], which will be discussed in Section IV, can be used to obtain accurate estimates of the entropy rate.

In our paper, we borrow the Proportional-Integral-Derivative (PID) controller, a generic feedback control loop mechanism widely used in industrial control systems, to dynamically alter the probing rate in order to stabilize the KGR to the user's requirement.

Our experimental results show that the adaptive channel probing system could adaptively change its probing rate due to noise, interference, channel impediments, user movement, and environment dynamics. Moreover, it could stabilize KGR by using the PID controller and satisfy the users' KGR requirement. If users want to generate a key fast, then the probing rate will be high but efficiency becomes low. In other words, channel efficiency depends on how fast the user wants to generate a key.

The contributions of our paper are:

- Mathematical model of the channel probing is built and the relationship between key generation and probing rate is derived.
- Desired key generation rate is satisfied by using a PID controller under different situations.

The rest of this paper is organized as follows. In Section II, we discuss the related works. Section III introduces the mathematical analysis of channel probing in shared secret key generation. Then, we detail the components of the adaptive probing system: Lempel-Ziv complexity and PID controller, in Section IV and Section V, respectively. We present the experimental results and analysis in Section VI. We conclude this paper and discuss future work in Section VII.

II. Related Work

There has been an increasing interest in exploiting the randomness and reciprocity of the wireless channel to generate shared secret keys between two parties [3], [5], [11]–[13]. Early research in this area focused on theoretical analysis [14]–[16], while most recent works are more interested in practical implementations of the key generation schemes using off-the-shelf wireless devices [2]–[4]. Previous work assumed an authenticated channel while generating shared secret keys [11]–[13]. One recent work removed this assumption and proposed a shared secret key generation algorithm using level-crossings and quantization to extract secret bits from an unauthenticated wireless channel [3]. Another work proposed a method for key generation based on phase reciprocity of frequency selective fading channels [17].

To the best of our knowledge, there is no previous work discussing the trade-off among the channel probing rate, key generation rate and bandwidth cost, or adaptively tuning the channel probing rate according to the channel dynamics introduced by the environment and user mobility. In this paper, we address these problems and build a system to achieve adaptive channel probing in real scenarios using off-the-shelf devices.

III. Channel Probing in Secret Key Generation

We introduce the process of generating shared secret key and measuring RSS in this section. We define the utility function and the KGR function. Then, we derive the relationship between utility, KGR and probing rate. Finally, we show how our adaptive probing system works.

A. Shared Secret Key Generation

In general, there are three steps to generate a shared secret key: advance distillation, information reconciliation, and privacy amplification [18]. First, advance distillation is used to collect information. This could be considered as two questions: what kind of information to collect and how to collect it. In our work, we extract the RSS from the wireless channel using off-the-shelf devices. A user sends a packet to the desired destination and waits for a reply. Both sender and receiver will receive a packet nearly at the same time and measure the RSS. Due to the principle of wireless reciprocity, the train of RSS measurements will have the same behavior on both sides. Second, information reconciliation is a form of error correction carried out between legitimate users in order to ensure the keys generated separately on both sides are identical. Last, privacy amplification is a method for reducing (and effectively eliminating) a third party's partial information about the legitimate key. This paper only focuses on the first step.

B. Received Signal Strength

In telecommunications, the received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal. It is often done in the intermediate frequency (IF) stage before the IF amplifier and can also be sampled by an internal Analog-to-Digital Converter (ADC). The 802.11 standard does not define any relationship between RSS value and power level in mW or dBm. Vendors provide their own accuracy, granularity, and range for the actual power (measured as mW or dBm) and their range of RSS values.

For an arbitrary time t, let $S(t)$ represent an analog continuous-time received signal strength, shown as a dotted line in Figure 1. The RSS value at any time t could be converted by ADC, denoted as $S_{ad}(t)$, shown as a solid line. Figure 1 shows an example of how ADC quantizes the analog signal strength, and we call the duration that the ADC converts a continuous signal to the same RSS value as *stagnant time*, denoted as s, such as the time between t_a and t_b. Stagnant time varies. Sometimes it is long, such as $t_o - t_n$, while other times, it could be very short, such as $t_j - t_i$. As the analog signal $S(t)$ could increase or decrease sharply, or it could also stay around a tiny range, stagnant time then could tend to infinity and also to zero.

To probe at each stagnant time, we are able to get only one non-duplicated RSS value no matter how many times

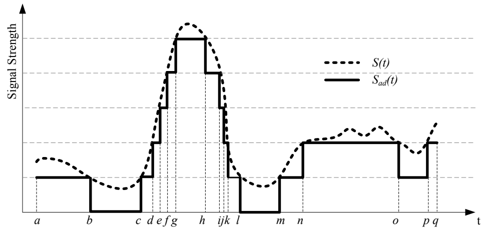

Fig. 1. Analog received signal strength with ADC

we probe. We call this RSS value the *effective RSS value*. The larger the sum of all effective RSS values, the more information we can extract from them.

C. Probing Process and Probing Sequence

The process of sending and receiving a probing packet pair, like ICMP PING and REPLY, is called a *probing process*. The time between two probing processes is called *probing interval*, or *interval*, denoted as θ. The larger the interval, the lower the *probing rate*, denoted as ν, where $\nu = 1/\theta$. A series of probing processes at the same interval is called a *probing sequence*.

If the interval is small, the probing process may happen more than once in a stagnant time that is larger than the interval, but only obtain an effective RSS value; we consider this case as *inefficient probing*. If the interval is large, the probing process may not occur in a stagnant time that is smaller than the interval; we call this case *inadequate probing*. If the interval is the same length as a series of equal-length stagnant times, we call this *perfect probing*. We could get a series of RSS values in which any two consecutive RSS values are not equal. However, in practice, a series of stagnant time will not be of exactly the same length, so perfect probing is hard. An optimal probing, which could obtain information from the channel as much as possible and also could probe in an efficient way, is the focus in this work.

D. Stagnant Time Distribution

For a given stochastic process $S(t)$ and non-constant function $S_{ad}(t)$, we have the discrete distribution $D(s_i)$ of stagnant times for $S_{ad}(t)$ shown as histogram in Figure 2, where $i \in N^+, s_{min} < s_i < s_{max}$, s_{min} and s_{max} are the minimum and maximum stagnant time, respectively. For an arbitrary value i, s_j is the next larger stagnant time after s_i, then the difference between s_i and s_j is Δs_i. We consider the sum of stagnant times equal to s_i is $D(s_i)\Delta s_i$. Therefore, the total number of all different stagnant times is $\sum_{i=min}^{max} D(s_i)\Delta s_i$. When i is an arbitrary value, if $\Delta s_i \to 0$, we have

$$\sum_{i=min}^{max} D(s_i)\Delta s_i = \int_{min}^{max} d(s)\mathrm{d}s, \tag{1}$$

where $d(s)$ is a fitted continuous curve, as solid line in Figure 2, from the discrete distribution $D(s_i)$, and $d(s) > 0, s_{min} \leq s \leq s_{max}$.

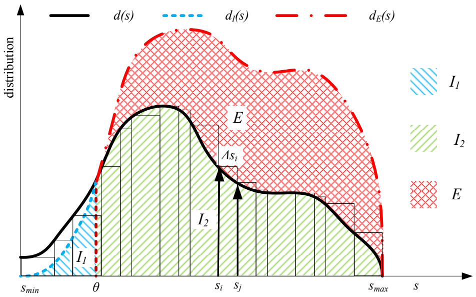

Fig. 2. Distribution functions of stagnant times

E. Functions and Properties

Suppose that the interval of a probing sequence is θ, and $0 < s_{min} < \theta < s_{max}$. For any $s > \theta$, this is an inefficient probing and we obtain a sum of effective RSS values, that is $I_2(\theta) = \int_{\theta}^{s_{max}} d(s)\mathrm{d}s$, where $I_2(\theta)$ is shown in Figure 2.

How many effective RSS values can we obtain from those stagnant time $s_{min} \leq s \leq \theta$? This is an inadequate probing and the probing process will miss some of the stagnant times. The smaller the stagnant time, the larger the missing probability. Therefore, the sum of effective RSS values we could obtain is $I_1(\theta) = \int_{s_{min}}^{\theta} d_I(s)\mathrm{d}s$, where $d_I(s) = d(s)\frac{s}{\theta}, s_{min} \leq s \leq \theta$ and $d_I(s)$ and $I_1(\theta)$ are shown in Figure 2. So, the total number of all effective RSS values is

$$I(\theta) = I_1(\theta) + I_2(\theta). \tag{2}$$

Since we can obtain more information from larger $I(\theta)$ values, we call $I(\theta)$ the *information function*.

When $s > \theta$, as an inefficient probing, some stagnant times will be probably probed more than once. The larger the stagnant time, the larger the re-probing probability. When re-probing happen at a stagnant time, only one RSS value is considered as effective, the others are called *ineffective RSS values*. The total number of all ineffective RSS values is

$$E(\theta) = \int_{\theta}^{s_{max}} (d_E(s) - d(s))\mathrm{d}s, \tag{3}$$

where $d_E(s) = d(s)\frac{s}{\theta}, \theta < s \leq s_{max}$. $d_E(s)$ and $E(\theta)$ are shown in Figure 2. Since the larger $E(\theta)$ is, the more inefficient probing becomes, we call $E(\theta)$ the *inefficiency function*.

Obviously, when the interval θ is getting larger, less effective RSS values will be obtained but the sum of ineffective RSS values decreases. The utility function is defined as

$$U(\theta) = \frac{I(\theta)}{E(\theta)}. \tag{4}$$

Lemma 1: When the probing interval becoming larger, information and inefficiency functions both decrease. But, the utility function increases.

Proof: According to Eq. 2 and Eq. 3, the derivatives of θ for functions $I(\theta)$ and $E(\theta)$ are

$$\begin{aligned} I'(\theta) &= [\tfrac{1}{\theta}\int_{s_{min}}^{\theta} d(s)s\mathrm{d}s]' + [\int_{\theta}^{s_{max}} d(s)\mathrm{d}s]' \\ &= -\tfrac{1}{\theta^2}\int_{s_{min}}^{\theta} d(s)s\mathrm{d}s \end{aligned} \tag{5}$$

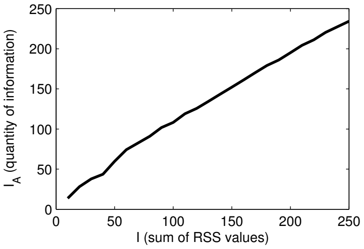

Fig. 3. Relationship between $I(\theta)$ and $I_A(\theta)$

$$E'(\theta) = [\int_{\theta}^{s_{max}} (d(s)\frac{s}{\theta} - d(s))\mathrm{d}s]' = -\frac{1}{\theta^2}\int_{\theta}^{s_{max}} d(s)s\mathrm{d}s. \tag{6}$$

As $0 < s_{min} < \theta < s_{max}$ and $d(s) > 0$, we have $I'(\theta) < 0, E'(\theta) < 0$. Therefore, information and inefficiency functions are both decreasing with θ.

The derivative of utility function $U(\theta)$ is

$$\begin{aligned} U'(\theta) &= \tfrac{1}{(E(\theta))^2}[I'(\theta)E(\theta) - I(\theta)E'(\theta)] \\ &= \tfrac{1}{[E(\theta)\theta]^2}[\textstyle\int_{\theta}^{s_{max}} d(s)\mathrm{d}s \int_{\theta}^{s_{max}} d(s)s\mathrm{d}s \\ &+ \textstyle\int_{\theta}^{s_{max}} d(s)\mathrm{d}s \int_{s_{min}}^{\theta} d(s)s\mathrm{d}s]. \end{aligned} \tag{7}$$

As $0 < s_{min} < \theta < s_{max}$ and $d(s) > 0$, we have $U'(\theta) > 0$. Therefore, utility function increases with θ. ■

Even if $0 < \theta \leq s_{min}$ or $\theta \geq s_{max}$, all lemmas are correct.

F. Key Generation Rate

We define the key generation rate as

$$K(\theta) = \frac{I_A(\theta)}{T}, \tag{8}$$

where $I_A(\theta)$ is the information estimation function based on Lempel-Ziv complexity and is proportional to $I(\theta)$, and T is the duration of probing sequence. The relationship between $I(\theta)$ and $I_A(\theta)$ is shown in Figure 3. Due to page limitations, further expositions and proofs are omitted. If a user's KGR requirement is κ, the PING interval should be $\theta = K^{-1}(\kappa)$, where $K^{-1}(\cdot)$ is the inverse function of $K(\cdot)$.

Lemma 2: When the interval θ becoming larger, the key generation rate decreases.

Proof: As derived in Lemma 1, $I(\cdot)$ is a decreasing function, and so is $I_A(\cdot)$. Therefore, $K(\theta) = I_A(\theta)/T$ is also a decreasing function. ■

Lemma 3: When KGR becoming larger, utility decreases.

Proof: When $K(\theta)$ increases, according to Eq. 8, we have $I_A(\theta)$ increasing and $I(\theta)$ increasing. As $I'(\theta) < 0$, in order to increase $I(\theta)$, we decrease θ. As $U'(\theta) > 0$, when θ decreases, we have $U(\theta)$ decreasing. ■

G. Adaptive Wireless Channel Probing System

In order to resolve $\theta = K^{-1}(\kappa)$, we introduce a PID controller to dynamically alter the PING interval and then to reduce the error between κ and actual KGR. Figure 4 represents a workflow of adaptive wireless channel probing system. After tuning parameters, such as the probing rate (i.e. time interval), the system starts to monitor the radio channel and one of the users then probes the channel by continually sending ICMP PING packets for a fixed duration, denoted as T_{ping}. Then legitimate users receive a series of PING packets and REPLY packets, respectively. RSS values are extracted, and then the entropy rate is estimated by LZ76 calculator and thereafter the KGR is calculated. Finally, the PID controller compares current loop KGR with desired KGR κ, then makes a new probing rate for next loop.

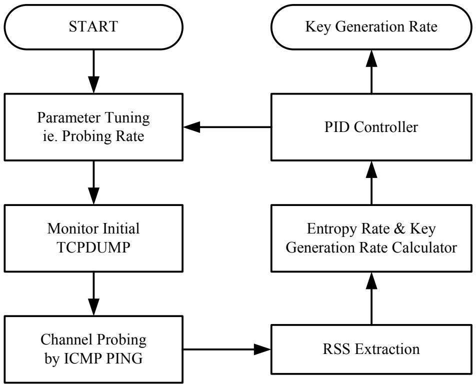

Fig. 4. Workflow of adaptive channel probing system

IV. Lempel-Ziv Complexity

In order to measure the quantity of information from a stochastic process, we give a brief introduction about entropy and entropy rate, which is practically estimated by Lempel-Ziv complexity. Then, the information estimation function $I_A(\theta)$ and a new KGR function are given.

A. Entropy and Entropy Rate

Let X be a random variable or random vector, taking values in an arbitrary finite set A, its *alphabet*, and with distribution probability $p(x)$ = Pr$\{X = x\}$ for $x \in A$. The *entropy* of X [8] is defined as,

$$H(X) = H(p) = -\sum_{x \in A} p(x) \log p(x). \tag{9}$$

The *entropy rate* H, or "per-symbol" entropy, of X is

$$H = H(x) = \lim_{n \to \infty} \frac{1}{n} H(X_1, X_2, \cdots, X_n), \tag{10}$$

whenever the limit exists, where $H(X_1, X_2, \cdots, X_n)$ is the entropy of the jointly distributed random variables $(X_1, X_2, \cdots, X_n)$.

B. Lempel-Ziv Complexity

We want to stress that the entropy is a property of sources and therefore difficult to evaluate [19]. In fact, the knowledge of the probability distribution involved in its calculation requires, in principle, an extensive sampling that usually cannot be performed, not to mention the reproducibility of the test conditions [20]. In contrast, the complexity as originally formulated by Lempel and Ziv (LZ76) [10] is a property of individual sequences that can be used to estimate the entropy.

Because of page limitations, we only give a brief introduction to show how LZ76 works. Any further properties and formal expression can be found in reference [10].

For a bitstring $X_N = [x_1, \cdots, x_N]$ of length N with $x_i \in \{0, 1\}$, a procedure that partitions X_N into non-overlapping substrings is called a *parsing*. A substring starting at position i and ending at position j of X_N which is the result of a parsing procedure is called a *phrase* $X_N(i, j)$. The set of phrases generated by a parsing of X_N is denoted with PX_N and the number of phrases $|PX_N|$ is denoted by q. Assume that a bitstring X_N has been parsed up to position i, so that $PX_N(1, i)$ is the set of phrases generated so far. The next phrase $X_N(i + 1, j)$ will be the first substring which is not yet an element of $PX_N(1, i)$. As an illustration, the string 0011001010100111 will be parsed as

$$0 \cdot 01 \cdot 10 \cdot 010 \cdot 10100 \cdot 111,$$

where $q = 6$.

In general, we define LZ76 value as

$$C_{LZ}(X_N) = \frac{q[\log_d q + 1]}{N}, \tag{11}$$

where d is diversity of samples in X or range of x, and

$$0 \leq C_{LZ}(X_N) \leq \log_2 d. \tag{12}$$

For a random sequence X_N from an ergodic and stationary source, entropy rate tends to [8], [21]

$$H = \lim_{N \to \infty} C_{LZ}(X_N). \tag{13}$$

In our paper, the RSS sequence is considered to be an ergodic and stationary source in a given time, like 1 second, if moving speed of user is not extremely high.

C. Information and KGR Function

During time T of a probing sequence, sum of the received RSS values is N, where $N = T/\theta$. We could estimate information by $I_A(\theta) = C_{LZ}(X_N)N = C_{LZ}(X_N)\frac{T}{\theta}$. Furthermore, from Eq. 8, we have detailed KGR function,

$$K(\theta) = \frac{I_A(\theta)}{T} = \frac{C_{LZ}(X_N)}{\theta}, \tag{14}$$

where $0 < \theta \leq \theta_{max}$, and θ_{max} will be discussed in Section VI.

V. PID Controller

Resolving $\theta = K^{-1}(\kappa)$ is critical. Unfortunately, an accurate relationship between κ and θ is not known in advance. Even though we take many tests to successfully get the function of $K^{-1}(\cdot)$, we will fail to resolve when users or other objects move, or when the radio environment varies. That is why we have to introduce feedback control to let the system reduce the error between actual KGR and desired KGR κ, also called *setpoint*, by adjusting the PING interval.

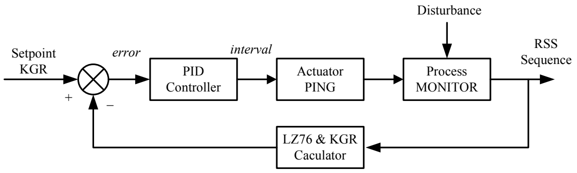

Fig. 5. Frame of PID control system

A. System Model

A series of probing processes with same interval is a probing sequence. We also call it a *loop* for the controller. In the ith loop, we set PING interval θ_i as input to probe channel. At the end of this loop, we get entropy rate $C_{LZ}(i)$ and key generation rate k_i as output and feedback to compare with κ. The PID controller then calculates a new interval θ_{i+1} for the next loop. The controller model is

$$\begin{aligned} \theta_{i+1} =& \theta_i + C_P(k_i - \kappa) \\ &+ C_I(\textstyle\sum_{N=i-\alpha}^{i}(k_i - \kappa)) + C_D(k_i - k_{i-1}), \end{aligned} \tag{15}$$

where $i = 1, 2, \cdots$, and α is the order of integral gain. C_P, C_I and C_D are proportional gain, integral gain and derivative gain, respectively. Figure 5 shows the frame of control system.

T_{ping} should be of appropriate duration. A large T_{ping} would decrease control performance while a small T_{ping} would decrease the stability of LZ76 calculator to estimate entropy rate. T_{ping} is a fixed parameter in our system, as 1 second. In order to keep the LZ76 calculator stable, we should limit the upper bound of θ, denoted as θ_{max}. As the limitation of hardware, we set the lower bound of θ at 1 ms. Thus, we have

$$1ms < \theta \leq \theta_{max}. \tag{16}$$

B. Stability

Define 1 (BIBO stability): BIBO stands for Bounded-Input Bounded-Output. If a system is BIBO stable, then the output will be bounded for every input to the system that is bounded.

Lemma 4: Our proposed PID control system is BIBO stable.

Proof: In our system, the interval is considered as input while KGR as output. Input θ is bounded in Eq. 16. Eq. 12 tells $C_{LZ}(X_N)$ is bounded between 0 and $\log_2 d$, and $K(\cdot)$ in Eq. 14 is bounded. Therefore, our system is BIBO stable. ■

C. Gain Parameters Tuning

The Ziegler-Nichols tuning method is a heuristic method of tuning a PID controller [22]. It is performed by setting the I and D gains to zero. The P gain is then increased (from zero) until it reaches the ultimate gain C_u, at which the output of the control loop oscillates with a constant amplitude. C_u and the oscillation period T_u are used to set the C_P, C_I, and C_D gains. They are $C_P = C_u/1.7, C_I = T_u/2, C_D = T_u/8$.

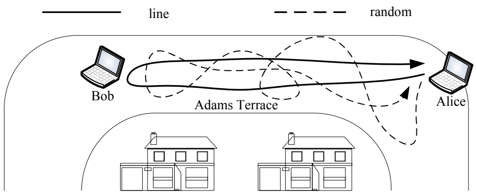

Fig. 6. Mobile type in Adams Community

VI. Experiment and Results

Our adaptive probing system runs on a platform that is composed of two DELL E5400 laptops with Intel WiFi Link 5300 802.11a/g/n wireless card. They both run a modified Fedora Linux kernel version 2.6.29-rc5-wl based on the wireless-testing tree. We made modifications to the Linux wireless device driver (iwlagn), the 802.11 stack (mac80211) and the kernel-to-userspace communication library (radiotap) for instrumentation purposes. The modifications allow the nodes to control the transmitter antenna and to record all three antenna RSS values per frame on frame reception. The RSS provided by the driver is an integer value in the range [-95,-20].

A. Experimental Setup

Outdoor and *Indoor*: The outdoor experiments are conducted at the Adams Terrace community in Davis, CA, USA. As shown in Figure 6, it is an open narrow straight road with several cars parked along the side and there are few people or cars moving along. The indoor experiments are conducted in a second floor bedroom of a townhouse.

Offline and *Online*: The procedure where laptops PING each other for a given time (60 seconds) at a constant interval without the PID controller is called the offline experiment, which is used to collect an RSS log and analyze the relationship between the interval and other metrics. The online experiment uses the PID controller to make KGR stable at setpoint, and logs operating parameters, which are used to analyze the performance of the system.

Static and *Mobile*, *Line* and *Random*: From Figure 6, we consider a static experiment if Alice and Bob are both fixed and no people or cars running through the road. We call it a mobile experiment if either one of them is moving. The mobile type includes line and random movements, shown as solid line and broken line, respectively.

The two laptops' transmission power are both set at 15 dBm. Moving speed is measured by a hand-held GPS.

B. Parameters: LZ76 Calculator

According to Eq. 11, Lempel-Ziv complexity of a finite sequence is determined by q, d, N. In a loop, q is calculated by a Python script after a finite sequence of RSS values. N is the length of a RSS sequence, that relate to the interval θ and duration time of a loop. d is a fixed number and is related to the diversity of RSS values. As our wireless card provides RSSI from -95 to -20 dBm, we consider diversity d as the total range, $d = 75$.

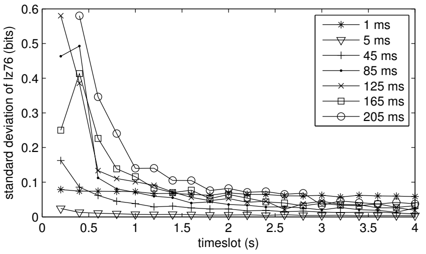

Fig. 7. Stand deviation of LZ76 calculator

TABLE I
Stand Deviation of LZ76 in 1 Second

Interval(ms)	**1**	**45**	**125**	**205**
Standard Deviation	0.0725	0.0380	0.1013	0.1401

As mentioned in Eq. 13, LZ76 is used to estimate entropy rate. If the sum of RSS values is not large enough, the LZ76 calculator will not be *stable*. Stable here means outputs of LZ76 calculator have a small variation.

We conduct a series of offline-outdoor-line-mobile experiments. PING interval θ is set as 5, 25,45, $\cdots$, 205 and 1 milliseconds (ms). After logging down all RSS into 12 files, we take one as an example to process data. As timestamps and RSS are both recorded, we select the RSS from the first timeslot of 200 ms as a group. The later RSS in the next timeslot of 200 ms as second group, and so on. We then calculate C_{LZ} of each group, and mean and standard deviation of those C_{LZ}. We then increase the timeslot from 200 ms to 400 ms. Next, we continue to increase timeslot, stepping at 200 ms, till 4 seconds. The same process is repeated on all the other log files.

Figure 7 shows standard deviation of C_{LZ} at different probing rates when timeslot increases from 200ms to 4s. Also shown in Table I, when timeslot is set as 1 second, standard deviations are all less than 0.15. As C_{LZ} in our experiments are mainly distributed from 0.6 to 1.2, standard deviation less than 0.15 could be considered as small enough. So, a timeslot of 1 second (i.e. $T_{ping} = 1$) and a PING interval of no more than 200ms (i.e. $\theta_{max} = 200ms$) could make LZ76 calculator stable.

C. Probing Rate vs LZ76

The relationship between LZ76 and probing rate is an important question. We agree with that a high probing rate would produce low LZ76, and vice versa. Here we adopt log files from the last experiment as Scenario I, and the timeslot is set as 1 second. The mean and standard deviation of C_{LZ} are drawn in Figure 8. In Scenario II, both laptops are static and separated away from each other about 30 meters. The PING interval is set as same as that in Scenario I. The mean of C_{LZ} in scenario II, shown in Figure 9, is not increasing as smoothly as the one in scenario I in Figure 8. Moreover, the C_{LZ} at any interval in Scenario I is larger than the one in scenario II. The C_{LZ} of static scenario could only rise to 0.79 at interval of

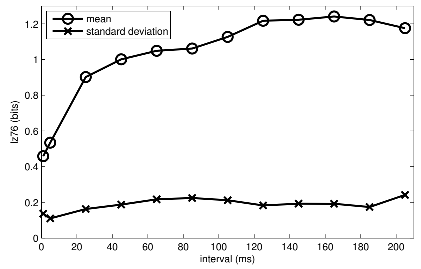

Fig. 8. LZ76 vs probing rate (Mobile)

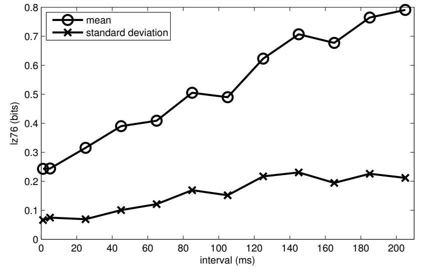

Fig. 9. LZ76 vs probing rate (Static)

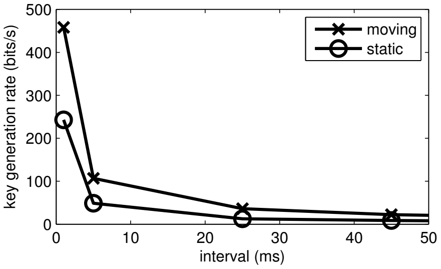

Fig. 10. KGR vs probing rate

205 ms, while mobile scenario reach 0.9 at interval of 25 ms. This result is reasonable. If two users are static, the channel is relatively stable. We are not able to obtain much randomness from this channel in a given time.

D. Probing Rate vs Key Generation Rate

Logs from previous offline experiments are analyzed in order to get the relationship between probing rate and KGR, which is calculated by Eq. 14. Figure 10 shows the results in mobile and static scenarios. At the same interval, the KGR is lower in the static scenario than that in the mobile scenario. To produce same KGR, it has to probe faster in the static scenario than in the mobile scenario. This indicates again that the users' movement increases the randomness of channel. Furthermore, the KGR in both scenarios decrease with interval θ, which has been derived by mathematical analysis in Lemma 2

E. Experimental Parameters: PID Controller

According to the Ziegler-Nichols method [22], the tuning parameters of PID controller are: $C_P = 0.0001, C_I = 0.000044, C_D = 0.000011$. The setpoint of the controller (i.e., desired KGR), is 50 bits/s and $T_{ping} = 1s$.

F. Metrics of Performance

Before listing metrics for online experiments, we introduce *Duplicated Index* (DI) to express the efficiency of the probing sequence. The larger the DI, the lower the efficiency. If we have a sequence like: ”AABBBCCCCC”, character ”A” has 1 duplicate and ineffective copy, and the weight of A over whole sequence is 2/10. The same process is repeated on the other characters. Thus, we have DI $= 1 \times \frac{2}{10} + 2 \times \frac{3}{10} + 4 \times \frac{5}{10} = 2.8$.

Here is the list of performance metrics studied:

- KGR mean error: $|\sum_{i=1}^{N} k_i/N - setpoint|$.
- KGR oscillation frequency: the times that k_i crosses through setpoint, denoted as N_{osc}, and oscillation frequency $f_{osc} = N_{osc}/N$.
- KGR overshoot: denote mean of overshoot as *overshoot-mean* and standard deviation of overshoot as *overshoot-std*.
- KGR settling time: when k_i first reach setpoint, consider the loop number as settling time.
- Ping interval: calculate mean and standard deviation of interval.
- Efficiency: duplicated index DI,

where k_i is key generation rate at ith loop, $i = 1, 2, \cdots, N$, and N is determined by online running time. All metrics above are used from Table II to Table IV.

G. Variable Motion

We conduct a series of online outdoor experiments with the PID controller. The first group of experiments shows how the interval varies when one user's moving speed changes from 0 m/s to about 1 m/s then back to 0 m/s within 90 seconds. As shown in Figure 11, at the beginning, users are both static and KGR is stabilized around 50 bits/s but with a very large overshoot. At about 32 seconds, one user starts to move. Suddenly, KGR increases sharply as a response, as movement introduces more randomness. Then, the PID controller makes the PING interval increasing in order to stabilize the KGR back to 50. At about 60 seconds, the mobile user stops. The KGR decreases and then the interval decreases. Results show that the KGR in the mobile phase seems more stable than in the static phase, and show that the PING interval is larger in the mobile phase than that in the static phase. The reason that the KGR overshoot in the mobile phase is much smaller than that in the static phase will be discussed in Section VII.

The second group of experiments shows how the user's moving speed affects system performance, shown in Table II. Setpoint is set at 50 bits/s. The mean errors of KGR in three different speeds are smaller than 0.3, we consider this as a contribution of PID controller. The faster the user moves, the smaller the oscillation frequency, and the smaller the overshoot. The most important results are that the faster the user moves, the larger the PING interval, and the larger the efficiency. Our adaptive probing system could adapt to speed variations; it decreases probing rate when the moving speed

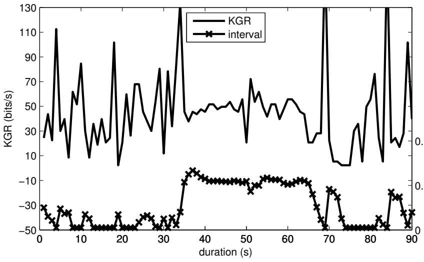

Fig. 11. Interval and KGR if speed vary

TABLE II
DIFFERENT SPEEDS

Moving Speed	0.3 m/s	0.8 m/s	1.5 m/s
Mean Error	0.0984	0.2426	0.1350
Oscillation Frequency	0.6000	0.5167	0.4583
Overshooting-mean	7.6733	6.0464	5.3605
Overshooting-std	8.1752	5.5041	4.6627
Settling Time (loop)	3	3	4
Ping Interval-mean	0.0171	0.0191	0.0282
Ping Interval-std	0.0036	0.0026	0.0022
Duplicated Index	1.7137	0.8969	0.6120

TABLE III
DIFFERENT MOBILE TYPES

Motion Type	Line	Random
Mean Error	0.0984	0.1496
Oscillation Frequency	0.6000	0.5167
Overshooting-mean	7.6733	7.3213
Overshooting-std	8.1752	11.5540
Settling Time (loop)	3	2
Ping Interval-mean	0.0171	0.0195
Ping Interval-std	0.0036	0.0037
Duplicated Index	1.7137	1.6496

rises. That is because the channel varies faster when the users move fast, so more random information is obtained.

The third group of experiments studies whether the type of movement affects performance. Line and random movements are drawn in Figure 6 and results are listed in Table III. The PING interval is larger in random type than in line type; this means the random mobile would extract more randomness information from wireless channel. Furthermore, random mobile has higher efficiency.

H. Different Sites

Another group of experiments are conducted to get the difference in performance between outdoor scenario and indoor scenario. Results are listed in Table IV, they show that the interval in indoor scenario is a little larger than that of outdoor scenario. This is caused by more complicated reflect and multipath effects in indoor scenarios, so the system can probe more slowly, with a higher efficiency.

TABLE IV
DIFFERENT SITES

Motion Type	outdoor	indoor
Mean Error	0.0984	0.8007
Oscillation Frequency	0.6000	0.5333
Overshooting-mean	7.6733	4.7323
Overshooting-std	8.1752	6.4082
Settling Time (loop)	3	3
Ping Interval-mean	0.0171	0.0212
Ping Interval-std	0.0036	0.0019
Duplicated Index	1.7137	1.5146

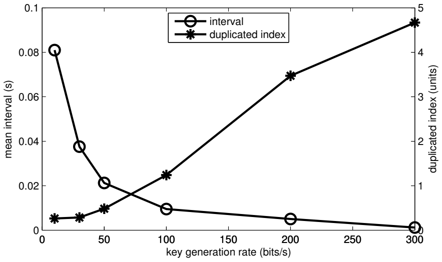

Fig. 12. Inteval and duplicated index in different KGR

I. Different Setpoint KGRs

The KGR, as setpoint in the PID controller, has been set at 50 bits/s in all previous experiments. This implies that we can generate a 50-bit key in 1 second. Obviously, the higher the setpoint, the faster we can generate a key, however, the lower the efficiency will be. This has been derived by mathematical analysis in Lemma 3. We conduct a new group online-moving experiment at home and set KGR at 10, 30, 50, 100, 200 and 300, respectively. What we are interested in is mainly how the interval and efficiency vary, shown in Figure 12. If we want to generate a key fast, then the probing rate will be high but efficiency become low, and vice versa. This tells users that if they want to use the channel efficiently, they should not set their KGR too high.

VII. CONCLUSION AND DISCUSSION

In order to satisfy users' requirement for key generation rate and to use the wireless channel in an efficient way, we introduce an adaptive channel probing system based on Lempel-Ziv complexity and PID controller. Theoretically, we build a mathematical model for channel probing and derive that the key generation rate (KGR) is proportional to probing rate. A utility function is also proposed and shows that the slower the probing rate, the higher the utility. However, too slow a probing rate is not acceptable by users who want to generate a key within a given time. In our paper, we avoid making an intractable decision between probing rate and efficiency. We instead consider satisfying the users' KGR as the primary goal. The PID controller is used to stabilize KGR as output according to input, such as PING interval.

A series of experiments are conducted to test performance in different speeds, different mobile types, different sites and different KGRs. Experimental results show that our channel probing system can adaptively change its probing rate due to noise, interference, other channel impediments, user movement and environment dynamics. It not only satisfies user's KGR requirement, but also makes the probing process as more efficient as possible.

However, from the experiments above, the overshoot of KGR seems a bit large. This may be as a result of three reasons. First, as the interval in the current loop is determined by KGR in last loop, and channel condition is not predictable. It is impossible to stabilize KGR exactly at setpoint. Second, the accuracy of the LZ76 calculator to estimate entropy rate is not high enough if the RSS sequence is not long enough. Extending PING time may improve the accuracy of LZ76 calculator. However, extending PING time may result in instability of the controller. Third, the parameters of PID controller may not be optimal.

Overshoot of KGR in Figure 11 tells a different problem. Larger overshoot in static phase is caused by the PID controller. In static phase, the interval is very small in order to satisfy desired KGR. For example, if the current KGR error is k, PID controller will subtract 1 ms from last interval to get a new interval. However, subtracting 1 ms from 2 ms in the static phase is very different from subtracting 1ms from 20 ms in the mobile static. This will cause large overshoot in static phase. Basically, that is because the control object is nonlinear but the controller is linear.

In order to solve the control problem mentioned above and improve the performance of system, we can use the adaptive controller to cope with the fact that the parameters of the system being controlled are slowly time-varying or uncertain, and this approach is considered our future work.

Acknowledgment

The authors would like to thank Lihua Dou, Daniel Wu, Jack(An) Chan, George Chen and Victor Omwando for discussions, and China Scholarship Council (No.2009603052) for support.

References

[1] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644 – 654, nov 1976.

[2] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2009, pp. 321–332.

[3] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2008, pp. 128–139.

[4] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *Proceedings IEEE INFOCOM*, 14-19 2010, pp. 1 –9.

[5] N. Patwari, J. Croft, S. Jana, and S. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 17 –30, Jan. 2010.

[6] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2001.

[7] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, 1948.

[8] J. Thomas, *Elements of Information Theory*. John Wiley and Sons, 1991.

[9] J.-L. Blanc, N. Schmidt, L. Bonnier, L. Pezard, and A. Lesne, "Quantifying neural correlations using lempel-ziv complexity," in *NEUROCOMP2008*, Marseille, France, 2008.

[10] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 75 – 81, Jan 1976.

[11] T. Aono, K. Higuchi, M. Taromaru, T. Ohira, and H. Sasaoka, "Wireless secret key generation exploiting the reactance-domain scalar response of multipath fading channels : RSSI interleaving scheme," *The European Conference on Wireless Technology*, pp. 173–176, October 2005.

[12] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 401–410.

[13] R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE International Conference on Ultra-Wideband*, pp. 270–275, September 2007.

[14] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[15] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*. Springer-Verlag New York Inc., 1994, pp. 410–423.

[16] C. Cachin and U. M. Maurer, "Linking information reconciliation and privacy amplification," *Journal of Cryptology*, vol. 10, pp. 97–110, 1997.

[17] H. Koorapaty, A. Hassan, and S. Chennakeshu, "Secure information transmission for mobile radio," *IEEE Communications Letters*, vol. 4, no. 2, pp. 52–55, Feburary 2000.

[18] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," *Journal of Cryptology*, vol. 5, pp. 3 – 28, 1992.

[19] S. P. Strong, R. Koberle, de Ruyter van Steveninck, and W. Bialek, "Entropy and information in neural spike trains," *Phys. Rev. Lett.*, vol. 80, 1998.

[20] J. M. Amigo, J. Szczepanski, ElekWajnryb, and M. V. Sanchez-Vives, "Estimating the entropy rate of spike trains via lempel-ziv complexity," *Neural Computation*, vol. 16, pp. 717–736, 2004.

[21] R. Badii and A. Politi, *Complexity: Hierarchical structures and scaling in physics*. Cambridge: Cambridge University Press, 1997.

[22] J. B. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *ASME Transactions*, vol. 64, pp. 759–768, 1942.