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: weiyunchuan1983@bit.edu.cn

†Department of Computer Science
University of California, Davis, CA 95616
Email: {ycwei,kaizeng,pmohapatra}@ucdavis.edu

***Abstract*—Generating a shared 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 bit generation rate (BGR) and channel resource consumption, nor adjust the probing rate according to different scenarios. In order to satisfy users' requirement for BGR and to use the wireless channel efficiently, we first build a mathematical model of channel probing and derive the relationship between BGR 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 uses 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 BGR under different moving speeds, different mobile types, and different sites. 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 bit generation rate at a desired value with mean error of less than 0.9 bits/s.**

I. Introduction

Generating a shared 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 the assumption of computational hardness. 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 between two parties from the wireless channel [2]–[5]. Two wireless entities exploit reciprocity, randomness, and location-specific properties of a wireless fading channel, and obtain highly correlated channel states and produce shared 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-theoretically secure, i.e., no matter how much computing resources the attacker has, the attacker cannot break the key even if it eavesdrops all the key generation messages exchanged between the two entities.

In recent implementations and experiments, the received signal strength (RSS) is widely used as the parameter from wireless channel to generate the shared key [3]–[5]. The RSS can be easily obtained from current wireless device drivers, so it makes key generation using off-the-shelf devices feasible. In order to generate a shared key, two parties need to send channel probing frames to each other to measure the RSS. We call this process *channel probing*. After channel probing, both parties quantize the measured RSS sequences into bits, and apply reconciliation methods to make the bits agree. Finally they apply privacy amplification method to discard the bit information revealed to an eavesdropper to generate a shared key.

As far as we know, all the existing key generation implementations probe the channel at a preset and constant rate without any consideration of channel variation or probing efficiency. On the one hand, if the channel does not change very frequently or drastically, even if a user could probe the channel at a high probing rate, it will get an RSS sequence with many consecutive duplicated RSS values. These duplicated RSS values do not contribute new bit information to the final key, thus, result in a low probing efficiency, i.e., redundant probings. On the other hand, it will take an intolerably long time to generate a key when probing at a very low rate.

The key generation rate (KGR) measures the number of shared secret bits generated per second between two parties. We call the bit information generated at each side per second

This research was supported in part by the US National Science Foundation through the grant CNS-0709264, the US Army Research Office through the MURI grant W911NF-07-1-0318, and the China Scholarship Council No.2009603052.

during the channel probing process as bit generation rate (BGR). Assume there is a relatively constant efficiency of the reconciliation and privacy amplification processes (i.e. the two processes discard a relatively constant portion of the bits generated at each side), then we can use BGR to infer KGR.

The BGR is largely determined by the channel variation and probing rate. More specifically, it is largely determined by the entropy rate of the RSS sequence and the time used to obtain the sequence. The entropy rate is, informally, the time density of the average information in a stochastic process [7]. In practice, since users always have requirements about how much time they can afford to generate a certain length of key, we could control the probing rate according to different channel variations to satisfy a certain BGR. That is, the system does not have to probe too fast to achieve a desirable BGR; only fast enough to avoid using the channel inefficiently.

In this work, we build a mathematical model of the channel probing system and derive the relationship between the BGR and probing rate, and use the entropy rate of the RSS sequence to estimate the BGR. 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 [8]. The concept of Lempel-Ziv complexity (LZ76) [9], which will be discussed in Section IV, can be used to obtain accurate estimation 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 tune the probing rate in order to stabilize the BGR according to the user's requirement under dynamic conditions.

Our experimental results show that the adaptive channel probing system can adaptively change its probing rate according to user movement and environment dynamics. Moreover, it can stabilize BGR by using the PID controller and satisfy the users' BGR requirement.

The contributions of our paper are:

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

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 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 keys between two parties [2]–[5], [10]–[16]. Early research in this area focused on theoretical analysis [13]–[15], 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 keys [10]–[12]. One recent work removed this assumption and proposed a shared 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 [16].

To the best of our knowledge, there is no previous work discussing the trade-off among the channel probing rate, bit generation rate and channel resource consumption, or adaptively tuning the channel probing rate according to the channel dynamics introduced by user mobility and/or the environment. 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 Shared Key Generation

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

A. Shared Key Generation

In general, there are three steps to generate a shared key: advance distillation, information reconciliation, and privacy amplification [17]. 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 sender transmits a request frame directly to a desired receiver and waits for a reply. The receiver instantly echoes a reply when it receives the request. Thus, both the sender and receiver will receive a frame nearly at the same time and measure the corresponding RSS. Due to the principle of wireless reciprocity, the train of RSS measurements will have the same characteristics (i.e., same variations) 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 (RSS) 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

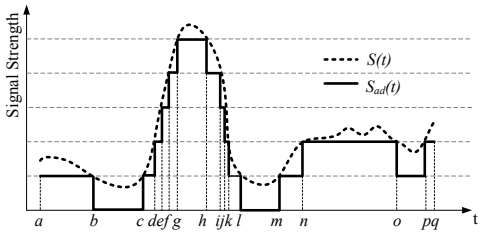

Fig. 1. Received signal strength and stagnant time

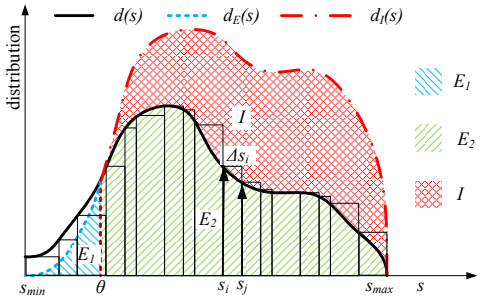

Fig. 2. Distribution functions of stagnant times

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.

In order to make our derivation more clear, for an arbitrary time t, let $S(t)$ represent an hypothetical analog continuous-time received signal strength, shown as a dotted line in Figure 1. Please note that $S(t)$ is only hypothetical and cannot be obtained by the off-the-shelf wifi device. If the channel probing is as fast as possible, the RSS value at nearly any time t could be converted by ADC, denoted as $S_{ad}(t)$, shown as a solid line. In contrast, $S_{ad}(t)$ is an actual measurement and can be obtained by off-the-shelf wifi device. As the solid line $S_{ad}(t)$ shows, we call the duration of having a consecutive equivalent RSS values sequence as *stagnant time*, denoted as s, such as the time between t_a and t_b. Stagnant time varies, from less than 1 millisecond, such as $t_j - t_i$, to even hundreds of milliseconds, such as $t_o - t_n$.

As the hypothetical analog signal $S(t)$ could increase or decrease sharply, or it could also stay around a tiny range, stagnant time then could tend to zero or to infinity.

To probe at each stagnant time, we are able to get only one non-duplicated RSS value no matter how many times we probe. We call this RSS value the *effective RSS value*. In a probed RSS sequence, small number of effective RSS values implies many duplications in the RSS measurements, which indicates a low probing efficiency.

C. Probing Process and Probing Sequence

The process of sending and receiving a probing packet pair, like ICMP PING and REPLY [18], 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* or a *loop* in online experiments with PID controller. Duration of a loop is denoted as T.

If the interval is small, the probing process may happen several times in a stagnant time that is larger than the interval, but only obtain one 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*. 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 weighted distribution $D(s_i)$ of stagnant times for $S_{ad}(t)$ shown as histogram in Figure 2, where $i \in \mathcal{N}^+, s_{min} \leq s_i \leq s_{max}$, 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.

Discrete weighted distribution $D(s_i)$ is the function of Δs_i, and $D(s_i)\Delta s_i$ equals to the total number of stagnant times that equal to 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

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

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

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 total number of effective RSS values, that is $E_2(\theta) = \int_{\theta}^{s_{max}} d(s)\mathrm{d}s$, where $E_2(\theta)$ 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 total number of effective RSS values we could obtain is $E_1(\theta) = \int_{s_{min}}^{\theta} d_E(s)\mathrm{d}s$, where $d_E(s) = \frac{s}{\theta}d(s), s_{min} \leq s \leq \theta$ and $d_E(s)$ and $E_1(\theta)$ are shown in Figure 2. Then the total number of all effective RSS values is

$$E(\theta) = E_1(\theta) + E_2(\theta). \tag{2}$$

More information obtained from the channel may result in larger $E(\theta)$ value.

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 happens in a stagnant time, only one RSS value is

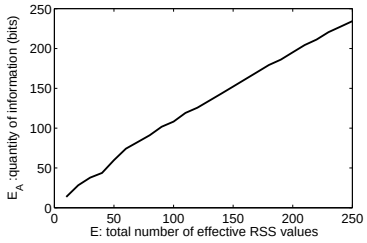

Fig. 3. Relationship between $E(\theta)$ and $E_A(\theta)$. This result is obtained from an experiment with θ as 10 ms under different durations.

considered as effective, the others are called *ineffective RSS values*. The total number of all ineffective RSS values is

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

where $d_I(s) = \frac{s}{\theta} d(s), \theta < s \leq s_{max}$. $d_I(s)$ and $I(\theta)$ are shown in Figure 2. The larger $I(\theta)$ is, the more inefficient probing becomes.

In addition, we define the utility function as

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

Lemma 1: When the probing interval becomes larger, $E(\theta)$ and $I(\theta)$ functions both decrease. However, the utility function increases.

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

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

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

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

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

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

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

Note that even if $0 < \theta \leq s_{min}$ or $\theta \geq s_{max}$, all lemmas in this paper are valid.

F. Bit Generation Rate

We define the bit generation rate as

$$B(\theta) = \frac{E_A(\theta)}{T}, \tag{8}$$

where $E_A(\theta)$ is the information estimation function based on Lempel-Ziv complexity (detailed in Section IV) and is proportional to $E(\theta)$, and T is the duration of a probing sequence. The relationship between $E(\theta)$ and $E_A(\theta)$ is shown in Figure 3. If a user's BGR requirement is β, the probing interval should be $\theta = B^{-1}(\beta)$, where $B^{-1}(\cdot)$ is the inverse function of $B(\cdot)$.

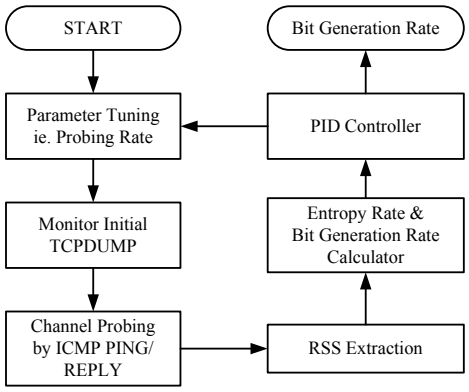

Fig. 4. Workflow of adaptive channel probing system

Lemma 2: When the interval θ becomes larger, the bit generation rate decreases.

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

Lemma 3: When BGR becomes larger, utility decreases.

Proof: When $B(\theta)$ increases, according to Eq. 8, we have $E_A(\theta)$ increasing and $E(\theta)$ increasing. As $E'(\theta) < 0$, in order to increase $E(\theta)$, 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 $\theta = B^{-1}(\beta)$, we introduce a PID controller to dynamically alter the probing interval and then to reduce the error between β and actual BGR, which will be detailed in Section V. Figure 4 represents a workflow of adaptive wireless channel probing system. After tuning parameters, such as the probing rate, the system starts to monitor the radio channel. Two parties probe the channel by continually exchanging ICMP PING and REPLY packets for a fixed duration, denoted as T. The RSS values of the probing packets are recorded, and the entropy rate is estimated by LZ76 calculator (detailed in Section IV) and thereafter the BGR is calculated. Finally, the PID controller compares BGR obtained in the current loop with a desired BGR β, then makes a new probing rate for the next loop.

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 $E_A(\theta)$ and a new BGR 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 [7] 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 rate is a property of a random process 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 [20]. In contrast, the complexity as originally formulated by Lempel and Ziv (LZ76) [9] is a property of individual sequences that can be used to estimate the entropy rate. 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 [9].

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. 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 [7], [21], entropy rate tends to

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

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

C. BGR Function

During time T of a probing sequence, the total number of the received RSS values is N, where $N = T/\theta$. We could estimate the information by $E_A(\theta) = C_{LZ}(X_N)N$. Furthermore, from Eq. 8 and Eq. 11, we have the BGR function as

$$B(\theta) = \frac{q(\log_d q + 1)}{T}, \tag{14}$$

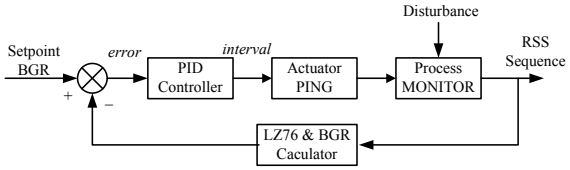

Fig. 5. Framework of the PID control system

where $0 < \theta \leq \theta_{max}$, and θ_{max} will be discussed in Section VI. Note that q is an unknown function of θ and stochastic channel variations.

V. PID Controller

Resolving $\theta = B^{-1}(\beta)$ is critical. Unfortunately, an accurate relationship between β and θ is not known in advance. Even though we take many tests to successfully get the function of $B^{-1}(\cdot)$, 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 BGR and a desired BGR β, also called *setpoint*, by adjusting the probing interval.

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 probing interval θ_i as input to probe channel. At the end of this loop, we get entropy rate $C_{LZ}(i)$ and bit generation rate b_i as output and feedback to compare with β. The PID controller then calculates a new interval θ_{i+1} for the next loop. The controller model is

$$\begin{aligned} \theta_{i+1} =& \theta_i + G_P(b_i - \beta) \\ &+ G_I \textstyle\sum_{j=i-\alpha}^{i} (b_j - \beta) + G_D(b_i - b_{i-1}), \end{aligned} \tag{15}$$

where $i = 1, 2, \cdots$, and α is the order of integral gain. G_P, G_I and G_D are proportional gain, integral gain and derivative gain, respectively. Figure 5 shows the framework of the control system.

Duration of a loop is T, and setting an appropriate T is very important. A large T would decrease control performance while a small T would decrease the stability of LZ76 calculator to estimate the entropy rate. T 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

Definition 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 BGR 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 $B(\cdot)$ 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 G_u, at which the output of the control loop oscillates with a constant amplitude. G_u and the oscillation period T_u are used to set the G_P, G_I, and G_D gains. They are $G_P = G_u/1.7, G_I = T_u/2, G_D = T_u/8$.

VI. Experiment and Results

Our adaptive probing system runs on a platform that is composed of two DELL E5400 laptops (called Alice and Bob, respectively) 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 fix the transmitter antenna and to record the 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 around. 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 duration (60 seconds in all experiments) at a constant interval without the PID controller, is called the offline experiment, which is used to collect RSS logs and analyze the relationship between the interval and other metrics. The online experiment uses the PID controller to make BGR stable at a setpoint, and, at the same time, logs necessary running parameters, which are used to analyze the performance of the system. All online experiments run 100 loops.

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 Bob 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. The moving speed is measured by a hand-held GPS.

B. Parameters: LZ76 Calculator (offline)

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 an RSS sequence, that relates 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 RSS from -95 to -20 dBm, we consider diversity d as the total range, $d = 75$.

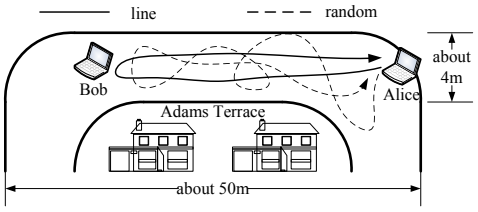

Fig. 6. Mobile type in Adams Community

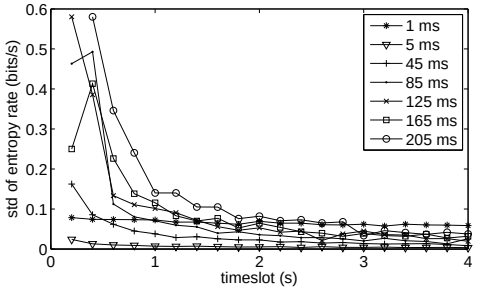

Fig. 7. Standard deviation of LZ76 calculator

From Eq. 13, LZ76 calculator cannot work well if N is not large enough. In order to make LZ76 calculator *stable*, we should carefully consider T and θ_{max}. Stable here means the outputs of LZ76 calculator have a small variation.

We conduct a series of offline-outdoor-line-mobile experiments. In the experiments, the probing interval θ is set as 1, 5, 25, 45, $\cdots$, 205 milliseconds (ms), respectively. After recording all the 12 RSS log files, for each log file, we process the data in the following way. We first truncate the items in the log file into groups for every 200ms timeslot according to timestamps. We then calculate the entropy rate of each group by LZ76 calculator, and also the mean and standard deviation. We then increase the timeslot from 200 ms to 400 ms, and do the same calculation. Next, we continue to increase timeslot, stepping at 200 ms, till 4 seconds. The same processing is repeated on all the log files.

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

C. Probing Rate vs Entropy Rate (offline)

We examine the relationship between entropy rate and probing rate under static and mobile scenarios in this subsection.

TABLE I
STD OF ENTROPY RATE IN 1 SECOND TIMESLOT

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

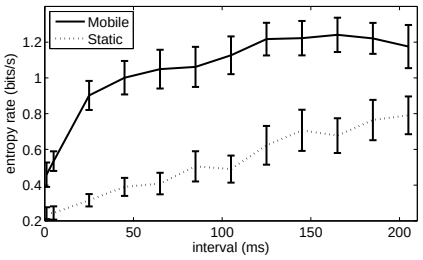

Fig. 8. Entropy rate vs probing rate (mobile and static)

According to Lemma 1 and Fig. 3, given a channel condition, a high probing rate (i.e., small probing interval) would produce low entropy rate, and vice versa. We adopt log files from the experiment in Section VI-B as *mobile* scenario, and the timeslot is set as 1 second. In *static* scenario, both laptops are static and separated away from each other about 40 meters. The probing intervals are the same as that in the mobile scenario. The mean and standard deviation of the entropy rates of these two scenarios under different probing intervals are drawn in Figure 8. We can see that, generally, the entropy rate is increased when the probing interval is increased in both scenarios. The entropy rate at any interval in the mobile scenario is larger than the one in the static scenario. The entropy rate of the static scenario could only rise to 0.79 at interval of 205 ms, while in mobile scenario it reaches 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 duration. While when the users are mobile, the channel is more variable, thus provide more randomness.

D. Probing Rate vs Bit Generation Rate (offline)

Logs from previous mobile and static offline experiments are analyzed in order to get the relationship between probing rate and BGR, which is calculated by Eq. 14. Figure 9 shows the results. At the same interval, the BGR is lower in the static scenario than that in the mobile scenario. To produce the same BGR, it has to probe faster in the static scenario than in the mobile scenario. This indicates the necessity of adaptively tuning the probing interval to achieve a desirable BGR under different scenarios. Furthermore, the BGR in both scenarios decrease with interval θ increased, which validated the mathematical analysis in Lemma 2.

E. Experimental Parameters: PID Controller

According to the Ziegler-Nichols method [22], the tuning parameters of PID controller are: $G_P = 0.0001, G_I = 0.000044, G_D = 0.000011$. The setpoint of the controller (i.e., desired BGR), is 50 bits/s and $T = 1s$. The order of integral gain $\alpha = 2$.

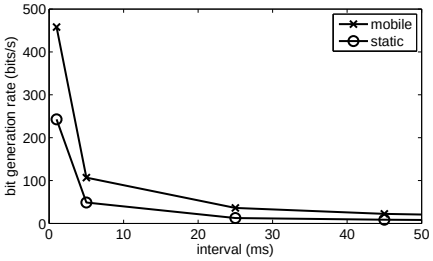

Fig. 9. BGR vs probing rate (mobile and static)

F. Metrics of Performance

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

Denote b_i as the bit generation rate at the ith loop, $i = 1, 2, \cdots, N$, and N is determined by online running time. Here is the list of performance metrics studied:

- BGR mean error: $|\sum_{i=1}^{N} b_i/N - \beta|$.
- BGR oscillation frequency (BGR Osc. Freq.): the times that b_i crosses through setpoint, denoted as N_{osc}, and oscillation frequency $f_{osc} = N_{osc}/N$.
- BGR overshooting (BGR Oversht.): overshoot refers to an output exceeding its final steady-state expected value.
- BGR settling time: when b_i first reaches setpoint, consider the loop number as settling time.
- Probing interval: the interval between two probing processes. It is dynamic in the online experiments, and we calculate the mean and standard deviation of it.
- Efficiency: duplicated index DI.

All metrics above are used from Table II to Table IV.

G. Variable Motion (online)

All the following online experiments setup are the same as the offline experiments, except that T is 1 second, setpoint β is 50 bits/s, $\theta_{min} = 1ms$ and $\theta_{max} = 200ms$, and each experiment runs 100 loops.

The first group of experiments shows how the interval varies when Bob'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 10, at the beginning, Alice and Bob are both static and BGR is stabilized around 50 bits/s but with a very large overshoot. At about 32 seconds, Bob starts to move. Suddenly, BGR increases sharply as a response, as movement introduces more randomness. Then, the PID controller makes the probing

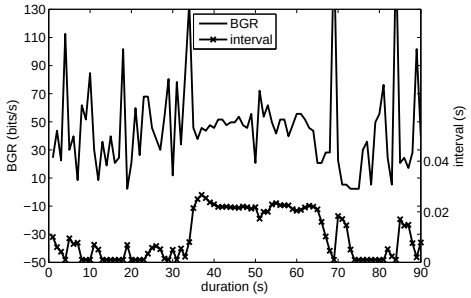

Fig. 10. Interval and BGR if speed vary

TABLE II
DIFFERENT SPEEDS

Moving Speed	0.3 m/s	0.8 m/s	1.5 m/s
BGR Mean Error	0.0984	0.2426	0.1350
BGR Osc. Freq.	0.6000	0.5167	0.4583
BGR Oversht. mean	7.6733	6.0464	5.3605
BGR Oversht. std	8.1752	5.5041	4.6627
Settling Time (loop)	3	3	4
Probing Interval-mean	0.0171	0.0191	0.0282
Probing 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
BGR Mean Error	0.0984	0.1496
BGR Osc. Freq.	0.6000	0.5167
BGR Oversht. mean	7.6733	7.3213
BGR Oversht. std	8.1752	11.5540
Settling Time (loop)	3	2
Probing Interval-mean	0.0171	0.0195
Probing Interval-std	0.0036	0.0037
Duplicated Index	1.7137	1.6496

TABLE IV
DIFFERENT SITES

Motion Type	outdoor	indoor
BGR Mean Error	0.0984	0.8007
BGR Osc. Freq.	0.6000	0.5333
BGR Oversht. mean	7.6733	4.7323
BGR Oversht. std	8.1752	6.4082
Settling Time (loop)	3	3
Probing Interval-mean	0.0171	0.0212
Probing Interval-std	0.0036	0.0019
Duplicated Index	1.7137	1.5146

interval increase in order to stabilize the BGR back to 50 bits/s. At about 60 seconds, Bob stops. The BGR decreases and then the probing interval decreases. It shows that the BGR in the mobile phase is more stable than that in the static phase, and the probing interval is larger in the mobile phase than that in the static phase. The reason why the BGR 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 Bob's moving speed affects the system performance, shown in Table II. The mean errors of BGR 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 probing interval, and the larger the efficiency. Our adaptive probing system can adapt to speed variations; it decreases probing rate when the moving speed rises.

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 probing interval is larger in the random type than in the line type; this means the random movement provides more randomness in the wireless channel. Furthermore, random movements has higher efficiency.

H. Different Sites (online)

Another group of experiments are conducted to get the difference in performance between outdoor scenario and indoor scenario. Bob is moving with random movement. 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 multi-path effects in indoor scenarios, so the system can probe more slowly, with a higher efficiency.

I. Different Setpoint BGRs (online)

The BGR, as setpoint β in the PID controller, has been set at 50 bits/s in all previous online experiments. 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-mobile-indoor experiment and set BGRs at 10, 30, 50, 100, 200 and 300 bits/s, respectively. The moving speed is about 0.3 m/s. Figure 11 shows the relationship between the probing interval and the efficiency (duplicated index). If we want to generate a key fast, then the probing rate will be high but the efficiency becomes low, and vice versa. It implies that if the users want to use the channel efficiently, they should not set their BGR too high.

VII. CONCLUSION AND DISCUSSION

In order to satisfy users' requirement for bit 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 bit generation rate (BGR) is proportional to probing rate. A utility function is also proposed and shows that the slower the probing rate, the higher the utility. However, a too slow probing rate is not acceptable by users who want to generate a key within a given time. In our paper, we

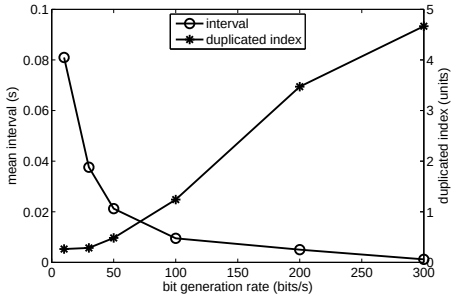

Fig. 11. Inteval and duplicated index in different BGR

avoid making an intractable decision between probing rate and efficiency. We instead consider satisfying the users' BGR as the primary goal. The PID controller is used to stabilize BGR 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 BGRs. Experimental results show that our channel probing system can adaptively change its probing rate according to user movement and environment dynamics. It not only satisfies user's BGR requirement, but also makes the probing process as efficient as possible.

However, from the experiments above, the overshoot of BGR seems a bit large. It may be due to the following three reasons. First, as the interval in the current loop is determined by BGR in the last loop, and channel condition is not predictable. It is impossible to stabilize BGR 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.

The overshoot of BGR in static phase in Figure 10 is caused by the PID controller. In static phase, the probing interval is very small in order to satisfy a desired BGR. If the current BGR error is k, PID controller will subtract 1ms from last interval to get a new interval. However, subtracting 1ms from 2ms in the static phase is very different from subtracting 1ms from 20 ms in the mobile phase. 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 the 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 as our future work.

ACKNOWLEDGMENT

The authors would like to thank Lihua Dou, Daniel Wu, An (Jack) Chan, George Chen, and Victor Omwando for discussions.

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, C. A., and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *Proceedings IEEE INFOCOM*, Mar. 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*. New Jersey,NJ,USA: Prentice Hall, 2001.

[7] J. Thomas, *Elements of information theory*. John Wiley and Sons, 1991.

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

[9] 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.

[10] 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, Oct. 2005.

[11] 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.

[12] 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, Sept. 2007.

[13] 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.

[14] 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.

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

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

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

[18] J. Postel, "Request for comments," RFC:792, Sept. 1981. [Online]. Available: http://tools.ietf.org/html/rfc792

[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,UK: 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.