Computer Networks 58 (2014) 112–126

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Multi-user wireless channel probing for shared key generation with a fuzzy controller

Lihua Dou [a], Yunchuan Wei [a,b,*], Jun Ni [c]

[a] Department of Automation, Beijing Institute of Technology, Beijing 100081, China
[b] Research and Development Center, China Academy of Launch Vehicle Technology, Beijing 100076, China
[c] College of Medicine, The University of Iowa, Iowa City, IA 52242, USA

ARTICLE INFO

Article history:
Received 9 November 2011
Received in revised form 5 March 2013
Accepted 23 August 2013
Available online 7 September 2013

Keywords:
Wireless channel probing
Shared key generation
Multi-user
Fuzzy controller
Information theory
Cryptography

ABSTRACT

Probing the wireless channel in wireless networks to generate a shared key is an increasingly interesting security topic. However, not much work has been focused on wireless channel probing in multi-user applications for Shared Key Generation (SKG). In this paper we propose a scheme of multi-user wireless channel probing using a broadcast approach and a fuzzy controller. In the proposed scheme, the concept of Desired-Weighting Factor (DWF) is introduced to meet a user's Key Generation Rate (KGR) requirement. The experimental results in this primary study show that the fuzzy controller can be used to satisfy KGR requirement by efficiently tuning the probing rate under dynamic conditions. Compared with the conventional Proportional–Integral–Derivative (PID) controller, the proposed probing scheme with a fuzzy controller may produce smaller overshoots and fewer oscillations. The fuzzy controller in the proposed scheme also stabilizes the KGR at desired values, improves the SKG accuracy, enhances the control capability, and increases the entropy rate. The study indicates that the proposed multi-user probing scheme can be used to make a trade-off between probing efficiency and the user's KGR requirement.

1. Introduction

Generating a shared key between two parties (legitimate users) through a wireless channel is an increasingly interesting topic of security in wireless networks [1–14]. The properties of a wireless probing channel such as reciprocity, randomness, and location–specification can be used to produce highly-correlated states in terms of bits for a shared key. Such key generation is information-theoretically secure, since a third party (an illegitimate user) half a wavelength away from the legitimate users may eavesdrop but has difficulty generating the same key in a rich scattering environment where the channel varies rapidly with time and spatial position [15]. The illegitimate user is unable to break down the key even with supercomputing power. One feasible and simple method to generate a shared key through a wireless channel is to use the Received Signal Strength (RSS) [1–4]. The RSS-based method has three major steps: quantization, information reconciliation, and privacy amplification. First, the legitimate users can send channel probing frames to each other to measure the RSS from participating wireless devices. This process is called channel probing. The measured RSS sequences can then be quantized in terms of bits. An information reconciliation process can be applied to correct the difference of bits streams obtained by different users in order to reach an agreement on the key. Often, a privacy-amplification process is then employed to remove unnecessary bits and to minimize the correlation between

* Corresponding author. Address: College of Medicine, The University of Iowa, Iowa City, IA 52242, USA. Tel.: +86 13810028259.
E-mail address: weiyunchuanmail@sina.com (Y. Wei).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.08.027

bits to make strong keys. The detailed steps can be found in [2] and the method implementations can be found in [2–5].

Early work concentrated on theoretical analysis [9–12,14], while recent work has focused on the implementations of Shared Key Generation(SKG) schemes using off-the-shelf wireless devices [1–4,13]. In practical implementations, the phase reciprocity of frequency selective fading channels [12,16] is employed.

Communications in many group-oriented multiple-user applications have various settings, ranging from multicasting in a network layer to teleconferencing/videoconferencing in an application layer. The privacy and integrity of such communications require specific security services. Although peer-to-peer security implementation has become more mature, the security of group communication remains challenging and relatively unexplored. People realize that SKG in group communication is not a simple extension of two-party communication. The demand for specific techniques to generate a group key in multi-user applications in group communication is relatively high [17,18].

As an extension of our previous work [1,34] that only concentrated on two-user application scenarios, this paper focuses on a wireless channel probing technique to generate a shared key for multiple users. The proposed multi-user channel probing scheme is based on broadcasting technology. Without losing generality, in this paper we consider three multi-user network topologies: single cluster, isolated multi-cluster, and networked multi-cluster. In order to meet multiple users' KGR requirement, we introduce an index called the Desired Weighting Factor (DWF). In addition, we employ a fuzzy controller to tune the probing rate for obtaining an actual KGR which is as close as possible to a desired value. We empower each user to have own desired KGR and corresponding DWF. Like the one-time pad cryptographic system, we try to increase the frequency of key changes in applications and provide a large desired-KGR value for high security consideration. In order to study the feasibility and applicability of the proposed SKG scheme on multi-user application scenarios, we experiment with different DWF values. For applications that either cannot tolerate any delay, any failure of key generation (e.g. videoconference), or are sensitive to the key generation rate, we set a large DWF value, while for ones that are not sensitive to the key generation rate we set a small DWF value.

In order to produce smaller overshoots and fewer oscillations, stabilize the KGR at desired values, improve the SKG accuracy, enhance the control capability, and increases the entropy rate, we use a fuzzy controller. The performance evaluation of the fuzzy controller is compared with the Proportional–Integral–Derivative (PID) controller in the proposed channel probing scheme.

The paper is organized as follows. Section 2 presents the system model, adversary model, and problem definition. Section 3 depicts the multi-user wireless channel probing scheme. Section 4 gives the test-bed setup. Section 5 presents experimental results and discussion, followed by a last section that gives conclusion with future work.

2. System model, adversary model and problem definition

2.1. System model

Assume there are N users in a wireless network, with each user considered as a node. Each user has equal communication and computation capacities. We term any two nodes within their communication coverage as a *pair*. Any pair holds a bidirectional wireless link. We consider two legitimate users (say Alice and Bob, as one of the pairs) who want to generate a shared key. Alice and Bob independently apply the following four steps [2]: channel probing, quantization, information reconciliation, and privacy amplification [19], respectively.

Channel probing is first used to collect wireless channel characteristics by legitimate users Alice and Bob. In this step, Alice and Bob exchange their request/reply probing frames within the duration T_p. Alice sends every probing request frame to Bob who instantly replies a packet back to Alice after he receives the request. We assume Alice sends the probing request frame with a fixed interval in single probing duration. At the end of the channel probing, Alice and Bob get channel measurements $\overrightarrow{H}_a$ and $\overrightarrow{H}_b$ respectively as

$$\begin{aligned}\overrightarrow{H}_a &= \{\hat{h}_a[1], \hat{h}_a[2], \hat{h}_a[3], \ldots, \hat{h}_a[N]\}^T \\ \overrightarrow{H}_b &= \{\hat{h}_b[1], \hat{h}_b[2], \hat{h}_b[3], \ldots, \hat{h}_b[N]\}^T\end{aligned} \tag{1}$$

where the superscript T denotes a matrix transpose and $\hat{h}_u[i]$ $(u = a, b; 1 \leqslant i \leqslant N)$ is the estimation of the channel characteristic $h_u[i]$ at time i. The subscript u stands for a user and $\overrightarrow{H}_a$ stands for the set of network channel characteristics in terms of a matrix or vector.

Quantization is used to convert the measured channel characteristics $\overrightarrow{H}_a$ and $\overrightarrow{H}_b$ into bit sequences. Information reconciliation is an error correction process carried out by both legitimate users in order to ensure that the keys generated separately on each side are identical [10]. During the reconciliation, some of the bitwise information may be revealed to an illegitimate user (Eve) from the eavesdropping during the communication between Alice and Bob.

Privacy amplification is a process to reduce or effectively eliminate Eve's partial information about the legitimate key and to minimize the correlation between the bits in a bit stream.

Any pair of legitimate network users would adopt these four processes. The details about how to implement the processes can be found in [2].

2.2. Adversary model

In an adversary model, we assume that there is an adversary (say Eve), who tries to break the key generation by eavesdropping on the communication among legitimate users. In this study, we make the following assumptions.

- Eve can read all the communication and can measure the channels.

- Eve knows in advance the key extraction algorithm and the parameters to be measured.
- Eve can be geographically close to (e.g. several wavelengths away), but not at, the location of the legitimate users.
- Eve can neither jam the communication channels nor modify messages. Eve cannot cause a man-in-the-middle attack. Eve's disruption of the key extraction and the authentication of legitimate users is not the focus of this paper.

2.3. Problem description

In multi-user networks, multi-pair channel probing can be achieved by a series of pair channel probing. Each pair probes the channel independently and simultaneously. Based on the probing independence, we could hypothesize that the adaptive scheme for SKG [1] could be used to improve the group-probing efficiency. This way the multi-user probing can be simply accomplished. Unfortunately, such a hypothesis may not be valid, since it focuses only on point-to-point communication within each pair and ignores group communication where multiple users cannot simultaneously launch probing processes because of co-channel interference.

For multi-user networks we propose a broadcast approach by sending a probing frame from a single user to multiple ones. When the receivers (legitimate users) obtain the frame, they instantaneously return messages to the sender as packets in sequence. The group broadcast probing rate can be determined by each user's desired-KGR and corresponding Desired Weighting Factor (DWF). In such group probing, a low broadcast probing rate may satisfy the requirements of users with a small desired-KGR but not satisfy the requirements of users who have a large desired-KGR. If the broadcast probing rate is high it meets most users' requirements, but decreases probing efficiency and wastes network and computational resources. A trade-off between all users' KGR requirements and probing efficiency is considered through a weighting factor described as follows.

3. Multi-user wireless channel probing

A key component in the proposed method is how to weight the importance of KGR requirements based on different network topologies. In this section, we consider the scales of the desired weighting factor based upon the user's requirements. In order to present the strategy of multi-user group channel probing, we classify various types of user clusters. We then present our methodology in terms of a strategic approach, the process scheme, and illustrative flowchart of our group channel probing.

3.1. Desired weighting factor

To best meet the users' KGR requirements while maintaining high probing efficiency, we introduce a Desired-key-generation-rate Weighting Factor (DWF). This factor helps determine the optimal probing rate with an acceptable efficiency. We classify the DWF into four scales.

- Trivial scale ($w = 0$): the KGR requirements are satisfied trivially by the user.
- Normal scale ($w = 1$): the KGR requirements are within the user's tolerance.
- Anxious scale ($w = 2$): the KGR requirements should be satisfied as much as possible.
- Essential scale ($w = 3$): the KGR requirements must be completely satisfied.

Different DWF values will be assigned to weight the group key generation rate (KGR) depending on each user's requirement satisfaction in our proposed group probing channel scheme. The values can leverage the important requirements by different users within the group communication.

3.2. Multi-user types

As long as there is a single node in a given network that can broadcast a probing frame to all the other nodes, we can assign the task of group probing to this node and obtain a group shared key. However, it is highly possible that no single node can broadcast a probing frame to all the other nodes. We identify three multi-user network types by their network topology: single cluster, isolated multi-cluster, and networked multi-cluster, respectively. The definitions of these types are as follows.

Type I: Single Cluster. A *single cluster* is a network group with N nodes and at least one single node is able to directly communicate with all other nodes. The group is considered to have a single *root node*, u_r, and $N-1$ *member nodes*, u_1, u_2, u_3, ..., u_{N-1} shown in Fig. 1a. This single cluster has N nodes and has $N-1$ pairs (connections). This type is an infrastructure employed for the rest of the network types. We may consider, for example, the root node as a base station and the member nodes as mobile users. Assume that each pair has a desired-KGR k_i that can be calculated based on channel measurements [1,34]. There are a total of $N-1$ pairs, and the system has $N-1$ desired-KGR values $k_1, k_2, k_3, \ldots, k_{N-1}$ and their corresponding DWF values, $w_1, w_2, w_3, \ldots, w_{N-1}$, respectively. The root node calculates the group broadcast probing rate in terms of newly-introduced parameters: weighed desired-KGR ($\mathrm{KGR_{wd}}$ or $\tilde{k}$) and weighed actual KGR ($\mathrm{KGR_{wa}}$ or $\widetilde{K}$. The weighed desired-KGR $\tilde{k}$ in Eq. (2) can be used to balance $N-1$ pairs' requirements.

$$\tilde{k} = \gamma \sum_{i=1}^{N-1} (k_i w_i) + \alpha \tag{2}$$

where γ denotes the group weighing coefficient and α denotes the group compensating value. Both γ and α will be discussed in Section 5.3. The weighed actual KGR $\widetilde{K}$ is the average of the actual KGR values. It can be calculated as in Eq. (3) and is used to process the actual KGRs from each pair.

$$\widetilde{K} = \frac{1}{N-1} \sum_{i=1}^{N-1} K_i \tag{3}$$

where K_i is the actual KGR between the root node u_r and the member node u_i.

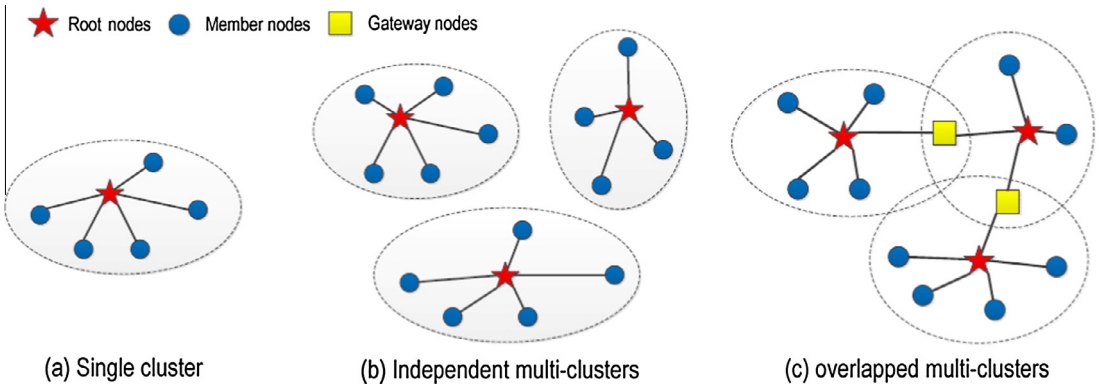


Fig. 1. Classification of multi-clusters with multiple nodes (users): (a) single cluster, (b) isolated multi-clusters, and (c) networked multi-clusters.

In general, the desired and actual KGR values are different. We can tune the broadcast probing rate and minimize the difference so the actual KGR can be as near as possible to the desired value. In order to achieve this goal, we employ a controller. In this paper we introduce a fuzzy controller, rather than the traditionally used PID controller. We compare these two controllers and present and discuss the experimental results in the following section.

Type II: Isolated Multi-Cluster. In this case, there are multiple clusters that communicate independently: the nodes within one cluster do not communicate with the nodes in another cluster. Such networks can be divided into several independent clusters, shown in Fig. 1b, with each of these being a single cluster. The probing scheme to be used is the same as given in Type I. Since these clusters are independent, nodes within a cluster do not suffer from channel conflicts.

Type III: Networked Multi-Cluster. In this case, there are also several clusters, but these clusters are not independent. At least one node in each cluster is connected with other node(s) in another cluster. In order words, different clusters share node(s) as shown in Fig. 1c.

The nodes belonging to more than one cluster are called *joint nodes*. There may be multiple joint nodes and the node chosen from among the joint nodes by an algorithm is defined as a *gateway node*. Therefore if a wireless network system has m multiple networked clusters, there should be m-1 gateway nodes.

Because of the gateway nodes, the root node of each cluster cannot simultaneously broadcast to all other member nodes. However, for Type II topologies one can design and implement a sequential process of channel probing for shared key generation, since each cluster only occupies the wireless channel for a certain period of time. This prompts us to analyze and allocate the time required for each cluster's group probing. We suggest that the time used to accomplish channel probing is proportional to computing the DWF. We define the total DWF for a cluster as the sum of the DWFs of each pair communication within the cluster. The total DWF becomes, therefore, the basis to determine how much time is required for the whole cluster's probing. The larger the total DWF, the longer the time a cluster consumes to accomplish the channel probing.

Type III clusters are more complex. Given m node-connected clusters in a Type III topology, with cluster j having S^j nodes. The total number of participating member nodes is $S^j - 1$. We let $w_i^j (i = 1, 2, 3, \ldots, S^j - 1)$ be the DWF value of the member node u_i^j in cluster j. The superscript j stands for the jth cluster (j = 1, 2, 3, ..., m) and the subscript i refers to the member node i. The summation of w_i^j, denoted as w_{sum}^j, is given by

$$w_{sum}^j = \sum_{i=1}^{S^j-1} w_i^j \tag{4}$$

Cluster j takes the following time percentage for its channel probing

$$\frac{w_{sum}^j}{\sum_{j=1}^{M} w_{sum}^j} = \frac{\sum_{i=1}^{S^j-1} w_i^j}{\sum_{j=1}^{M} w_{sum}^j} \tag{5}$$

This method can also be applied to both Type I and Type II topologies. For Type I (m = 1), the time percentage is unity. For Type II, the process of channel probing can be in either sequential or parallel, and the communication mode can be broadcasting within each cluster as in Type I. Once it takes over the wireless channel, each cluster uses exactly the same probing process as described in Type I. The channel probing procedure can be given in the following scheme.

3.3. Multi-user wireless channel probing scheme

A detailed description of the channel probing scheme in a single cluster is given in Fig. 2. First, the root node initializes parameters, obtains desired-KGR values k_i and calculates the weighed desired-KGR $\tilde{k}$ from Eq. (2). Monitoring the radio channels for all the participating nodes can be done by open source tools such as tcpdump [20] (a common packet analyzer) or the monitoring software pcap to capture packets traveling over a network. For UNIX or LINUX platforms, pcap can be found in the libpcap library; for Windows systems, WinPcap [20] can be used. The GUI network packets analyzer Wireshark [21], based on the open source pcap/tcpdump protocol, could be used as well.

All these software tools allow the users to intercept and display TCP/IP and other packets being transmitted or received. The root node probes the channel by continually

broadcasting ICMP PING and receiving REPLY packets [20], while the member nodes probe the channel by returning REPLY packets and receiving PING packets.

As a pair, the root and member nodes receive a frame nearly at the same time instant. Thus the pair based corresponding RSSs can be measured. Due to wireless reciprocity, the RSS measurements have the same characteristics (i.e., same physical quantities) at both root and member nodes. Next, all the nodes within a cluster estimate its entropy rate as the index of efficiency. After information reconciliation and privacy amplification, each legitimate user calculates the actual KGR. The last step is to reduce the difference between the desired and actual KGR values. The root node calculates the weighed actual KGR $\tilde{K}$ and compares it with the weighed desired-KGR $\tilde{k}$. A fuzzy controller can be introduced to figure out the new probing rate for the next loop during an iterative process.

The process of sending and receiving a probing packet pair, like ICMP PING and REPLY, is called a *probing process*. The time of a probing process is measured in terms of the *probing rate f* (unit: Hz). The *probing duration* is denoted as T_p. A series of probing processes at the same probing rate is called a *probing loop*.

3.4. Lempel–Ziv complexity and entropy

In order to measure the quantity of information from a stochastic process, we need to compute the entropy and entropy rate. For example, to estimate the probing efficiency, we need to calculate the entropy rate of the RSS sequence as the index. Intuitively, a high probing rate results in a low entropy rate with large KGR.

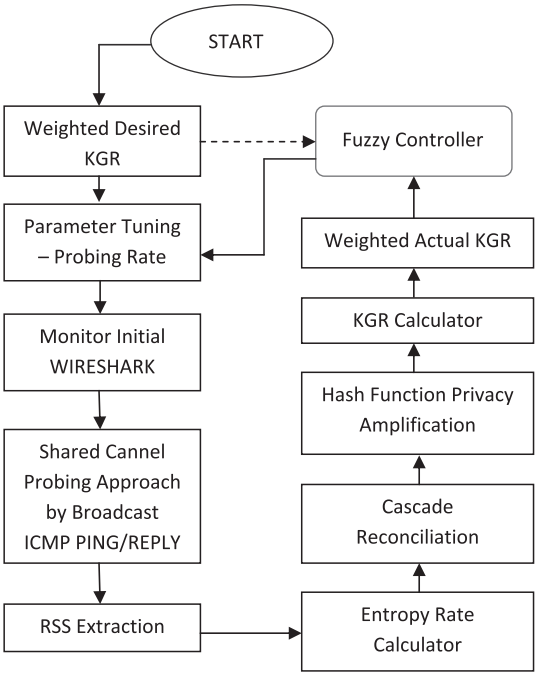


Fig. 2. Flowchart of multi-user wireless channel probing for a single cluster.

The entropy rate is a property of a random process and is therefore difficult to evaluate [24]. In fact, the knowledge of the probability distribution involved in its calculation requires, in principle, an extensive sampling that usually cannot be performed [25]. In contrast, the complexity as originally formulated by Lempel and Ziv (LZ76) in 1976 [22] 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. Detailed expression can be found in [22].

Let X be a random variable or random vector, taking values in an arbitrary finite set A and with distribution probability $p(x) = \Pr\{X = x\}$ for $x \in A$. The *entropy* of X is expressed as [23]

$$H(X) = H(p) = -\sum_{x \in A} p(x) \log p(x) \tag{6}$$

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

$$\lim_{n \to \infty} \frac{1}{n} H(X_1, X_2, X_3, \ldots, X_n) \tag{7}$$

whenever the limit exists. Here $H(X_1, X_2, X_3, \ldots, X_n)$ is the entropy of the jointly distributed random variables $X_1, X_2, X_3, \ldots, X_n$, and n is the total number of random variables.

The equation that estimates the *entropy rate* by Lempel–Ziv complexity (LZ76) [22–24] can be expressed as

$$C_{LZ}(X_N) = \frac{q(\log_d q + 1)}{n} \tag{8}$$

where subscript d is the diversity of samples in x range, and N is the total number of RSS values. The detailed discussions about the parameter q can be found in [1,34,22–24].

3.5. Feedback controllers

In order to calculate an "error" value as the difference between a measured process variable and a desired setpoint, a controller (algorithm) in the field of process and control engineering [28] can be employed. In this study, in order to produce smaller overshoots and fewer oscillations, to stabilize the KGR at desired values, to improve the SKG's accuracy, to enhance the control capability, and to increase the entropy rate, we deploy a process controller to obtain RSS sequences for KGR extraction in channel probing. In the proposed channel probing scheme, we will use a fuzzy controller in this study and compare its performance results with the results by using a traditional Proportional–Integral–Derivative (PID) controller. Both the fuzzy controller and PID controllers are well established. Due to page limitations, we only briefly address their mechanisms. Details can be found in [28].

Proportional–Integral–Derivative (PID) controller. The PID controller has a generic control loop feedback mechanism that provides control actions for specific process requirements. The controller attempts to minimize the error by tuning the process control inputs (parameters). The response of the controller can be an error, the degree to which the controller overshoots the setpoint and the degree of system oscillation. The PID controller is fully adequate for some applications that commonly feature low

to moderate control-quality (e.g. timing or precision) constraints and well-defined and stable dynamic system behavior. The PID controller has been widely used in industrial control systems or process engineering.

The PID controller can be deployed to obtain KGR sequences in the present study of channel probing. The framework with a PID controller in the RSS-KGR control process is shown in Fig. 3. In the ith loop of process control, we set the probing rate f_i as input to probe channel. At the end of this loop, we get the KGR value.

K_i as output actual KGR and as feedback to be compared with the desired KGR k. The PID controller then calculates a new probing rate f_{i+1} for the next loop. The controller model is

$$f_{i+1} = f_i + G_p(K_i - k) + G_I \sum_{j=i-\alpha}^{i} (K_j - k) + G_D(K_i - K_{i-1}) \qquad (7)$$

where $i = 1, 2, 3, \ldots$, and α is the order of integral gain. The values G_p, G_I and G_D are proportional, integral and derivative gains in the PID model [1,34], respectively.

Fuzzy controller. Since the concept of fuzzy logic was coined by Lotfy Zadeh in 1965, it has become an important mathematical and engineering tool to deal with uncertain, imprecise, or qualitative decision-making problems. A fuzzy logic based control system, called a fuzzy logic controller (or simply a fuzzy controller) is a mathematical system that analyzes analog inputs in binary bits. Fuzzy controllers that combine intelligent and conventional techniques are commonly used in the intelligent control of complex dynamic systems.

The principle of fuzzy controllers is conceptually very simple. The controller consists of an input, a processing, and an output stages. The input stage maps sensor or other inputs to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value.

One benefit of using fuzzy control is that it can handle a large number of inputs, most of which are relevant only for some special conditions. Such inputs are activated only when the related condition prevails. In this way, little additional computational overhead is required for adding extra rules. As a result, the rule-based structure remains understandable, leading to efficient coding and high system performance. There are a large number of successful applications of using fuzzy controllers. The details about theoretical control modeling and implementations of fuzzy controllers can be found in [26–28].

People use both fuzzy controllers and traditional PID controllers. Their performances are compared from case to case. For example, they have been designed and compared in on-line control of an advanced biodiesel microwave reactor system in a trans-esterification chemical process [29]. Experimental results indicate that the PID controller with good tuning gives good performance, while fuzzy controller has minimum overshoot, undershoot, and steady state error in temperature control, which is its most important parameter in biodiesel production. They are designed and compared in transportation studies [30] and fuzzy controllers are shown to be more robust: they are easy to use with existing models, and can easily deal with more inputs/outputs in process control systems.

Overall, PID controllers are widely applied in industrial processes owing to their simplicity and effectiveness for both linear and nonlinear systems, while fuzzy controllers have fairly high performance and are more robust in handling large numbers of controlling variables, especially working with computing power.

Recently, people have developed combined controllers such as fuzzy PID and fuzzy PI controllers [31,32]. However, these innovative controllers technically remain open and there is still a need to study the controllers' easy-tuning feasibility and performance applicability.

In this study, we propose to implement a fuzzy controller [26–28] to determine the probing rate. The architecture of the fuzzy controller is given in Fig. 4.

The fuzzy controller is composed of four basic components: fuzzification, rule base, inference mechanism, and defuzzification. We calculate the error $e(i)$ between $\tilde{k}$ and $\widetilde{K}$ and its differential error $e'(i)$ as the inputs to the controller, and the probing rate $f(i)$ as the output from the controller. The design of our controller is briefly given as follows.

Membership functions. In the controller, all real variables are first fuzzificated into fuzzy variables. As usual, the fuzzy variables are defined as *Negative Large* (NL), *Negative Middle* (NM), *Negative Small* (NS), *Zero* (ZO), *Positive Small* (PS), *Positive Middle* (PM) and *Positive Large* (PL), respectively. Each fuzzy variable has its own triangular membership function μ, shown in Fig. 5. Input sets include $e(i)$ and $e'(i)$, shown in Fig. 5a and b, respectively. The rules are listed in Table 1.

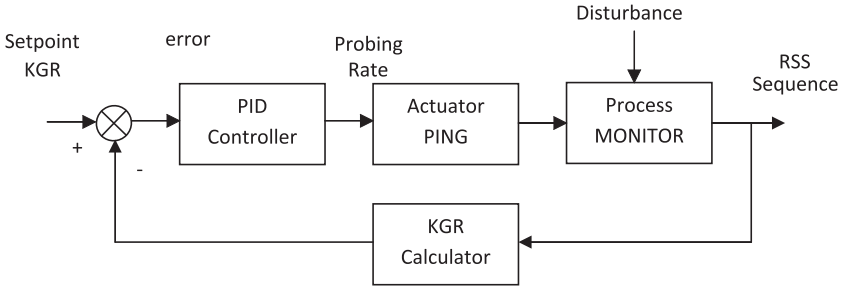


Fig. 3. Framework of the PID controller in RSS-KGR control process.

Inference mechanism. For any inputs $e(i)$ and $e'(i)$, each of them has two fuzzy variables (sometimes they are the same) according to the membership functions. For example, when $e(i) = 6$; $e'(i) = -1$, the fuzzy values of $e(i)$ are "PS" and "PM" while those of $e'(i)$ are "ZO" and "NS" according to Fig. 5. Thus, in the maximum condition four rules are matched, such as "NS", "NM", "ZO", and "NS" as in Table 1.

Defuzzification. In most situations, multiple rules are inferred and applied. The center of gravity (COG) defuzzification method can be employed for combining the recommendations represented by the implied fuzzy sets from all the rules [22]. According to the COG method, we have $F(i) = -1$ as the output. As the weighed actual KGR is larger than the weighed desired-KGR ($e > 0$) and the error is reduced ($e' < 0$), the system needs to decrease the probing rate ($F' < 0$).

4. Experimental setup

The testbed of our multi-user wireless channel probing system is composed of four Gateway LT25 laptops (users or nodes, called Alice, Bob, Carol, and Eve, respectively). Each has an Atheros-AR-5B95 802.11a/g/n wireless card. All use the Fedora Linux operating system with kernel version 2.6.34.8-68.fc13.i686.

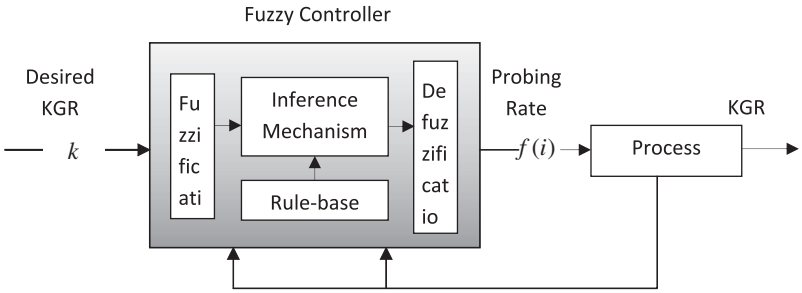


Fig. 4. Architecture of fuzzy controller.

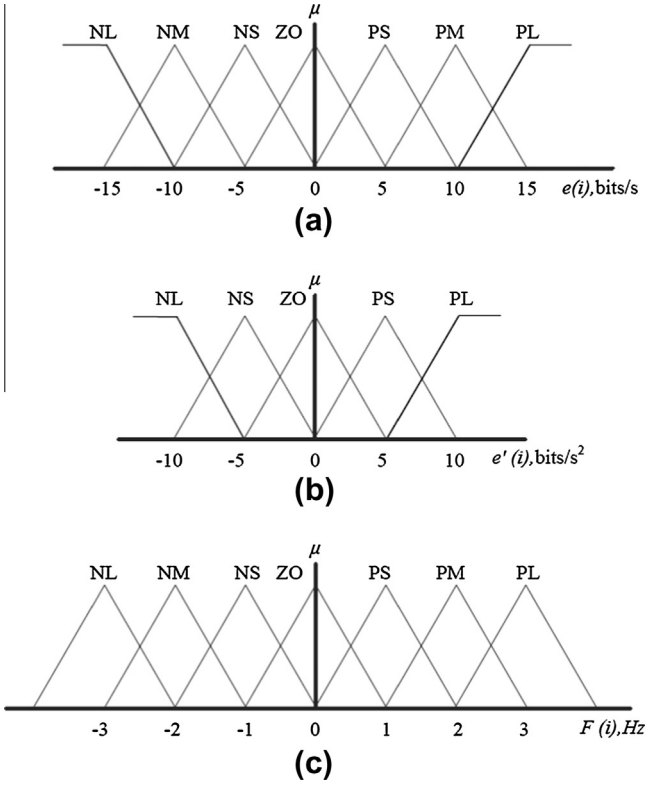


Fig. 5. Membership functions.

Table 1
Rules Table.

Incremental probing rate		KGR Error f_1						
		PL	PM	PS	ZO	NS	NM	NL
KGR differential error f_2	PL	NL	NL	NL	NM	NS	ZO	PS
	PS	NL	NL	NM	NS	ZO	PS	PM
	ZO	NL	NM	NS	ZO	PS	PM	PL
	NS	NM	NS	ZO	PS	PM	PL	PL
	NL	NS	ZO	PS	PM	PL	PL	PL

4.1. Experimental scenarios

There are two groups of experiments. The first evaluates the performance of the fuzzy controller as compared with the results using the PID controller. The experiment has two laptops (Alice and Bob). The second is to implement the multi-user scenario by using three laptops. Detailed scenario configurations are described as follows.

Outdoor and indoor. Our outdoor experiments were conducted at a public parking lot on the campus of Beijing Institute of Technology. The indoor experiments were conducted inside a campus building. Fig. 6 is an illustration of these scenarios.

Static and mobile (line and random). A static scenario is defined as having the users Alice, Bob, and Carol all stationary. There are no other moving objects. If moving objects are allowed, we call it a mobile scenario. The types of object motion include straight-line and random movements. In a two-user mobile scenario, we assume Alice is the moving object and Bob is still. In a three-user mobile scenario, we assume that Alice is moving and the other two are stationary. The transmission power of all the laptops was set at 20 dBm. The velocities of the objects (users) were measured by a hand-held GPS.

The weather was sunny when all the experiments were conducted. The outdoor temperature and humidity were about 27 °C and 55%, respectively, while the indoor temperature and humidity were 23 °C and 48%.

4.2. Performance indices of controller

The fuzzy controller in this probing scheme was specially designed. A series of experiments involving two users were conducted to test the performance of the fuzzy controller. The performance results were compared with the ones using a PID controller [34] in our previous study. Denote by $K_i (i = 1, 2, 3, \ldots, M)$ the actual KGR at the ith loop. M is the maximum number of loops determined by the probing duration. The following is a list of our test performance procedures:

KGR – mean: $v_m = \sum_{i=1}^{M} \frac{K_i}{M}$.
KGR – std: standard deviation of KGR, $\sqrt{\frac{1}{M}\sum_{i=1}^{M}(K_i - v_m)^2}$.
KGR – error: $e(i) = |\sum_{i=1}^{M} \frac{K_i}{M} - k|$.
KGR Oscillation frequency (KGR Osc Freq.): $\frac{M_{osc}}{M}$, with M_{osc} as the times that K crosses through the setpoint.
KGR overshoot (KGR Oversht.): the amount that K exceeds its designed value k.

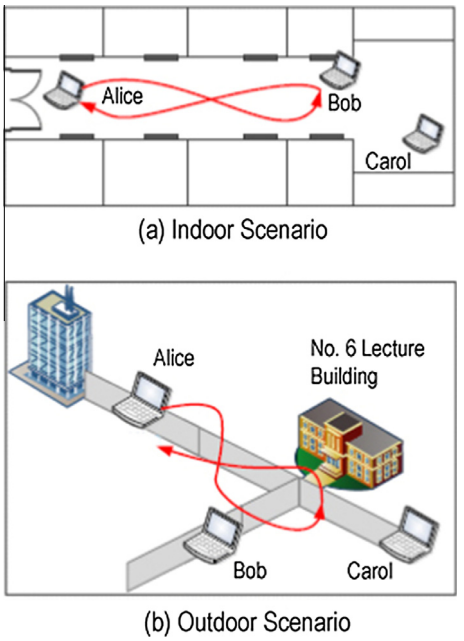


Fig. 6. Experimental Scenarios.

KGR settling time: the time for K to reach the setpoint the first time, taking the loop number as settling time.
ITAE (Integral of Time and Absolute Error): $\frac{\sum_{i=1}^{M} e(i)M}{1000}$.
Efficiency: entropy rate estimated by LZ76.

The KGR oscillation frequency, overshoot, and settling time jointly determine the ITAE. The smaller the ITAE, the better the controller works.

4.3. Parameters used in fuzzy controller

The parameters used in the fuzzy controller can be extracted from the membership functions shown in Fig. 5. For the sake of the controller's stability, we limit the range of the probing rate f from 5 to 300 Hz. We set the probing duration $T_p = 1$ s.

The parameters used in the PID controller are relative optimal and can be found in Section V of Refs. [1,34]. The value of the integral gain is $\alpha = 2$. The proportional, integral, and derivative gains are $G_P = 0.41$, $G_I = 0.23$, and $G_D = 0.16$, respectively.

4.4. Example: three-user probing

In order to depict the multi-user probing method, we used a three-user scenario (Alice, Bob and Carol). This can easily be extended to scenarios involving more users. As shown in Fig. 7, this scenario belongs to the single cluster type (Type I). Alice was selected as the root node and there were two pairs. The desired-KGR of the Alice–Bob pair was $k_{ab} = 75$ bits/s, and the desired-KGR of the Alice–Carol pair was $k_{ac} = 10$ bits/s. Alice tuned its broadcast probing rate by comparing the weighed desired-KGR $\tilde{k}$

with the weighed actual KGR $\widetilde{K}$. We set the parameters $\gamma = 0.5$ and $\alpha = 5$ in Eq. (2), respectively.

4.5. Cascade reconciliation and privacy amplification

When using cascade reconciliation [10], Alice permutes the bit stream randomly and divides it into small blocks. Alice then sends the block of permutation and parity information sequentially to Bob. Bob permutes his bit stream randomly in the same way, also divides it into small blocks, and calculates and checks if the parity of the blocks are identical to the one received from Alice. For each block whose parity does not match, Bob performs a binary search to find out whether a small number of bits in the block can be modified to match the parity information. These steps are iterated until the probability of successful match becomes higher than a desired threshold. Since information reconciliation is a probabilistic technique, it may fail occasionally. In case of such failure, the bit streams can be discarded and the key extraction process can be restarted by measuring the RSS values again. However, a low failure probability can be achieved by choosing the number of passes and the block size appropriately. In this study, we set the number of passes as 7 and the block size as 8, which can achieve an acceptable low failure probability.

Privacy amplification is another subsidiary but important process. It helps reduce the size of the output in a bit stream and eliminates the defects caused by the correlation in bits and the revealed information. In this study, privacy amplification is accomplished by letting both Alice and Bob use universal hash functions to obtain smaller output length from longer input streams. The Merkle–Damgard hash function (a collision-resistant one-way compression function) was used in our experiments [33]. This breaks the input bit stream into small blocks with a fixed size (5 bits). The bits in each block are hashed into 4 bits with average compression ratio 0.8. According to the statistical analysis of cascade reconciliation, the compression ratio is qualified to amplify privacy even though there are some bits leaking to Eve and some bits lead to reconciling mismatched bits.

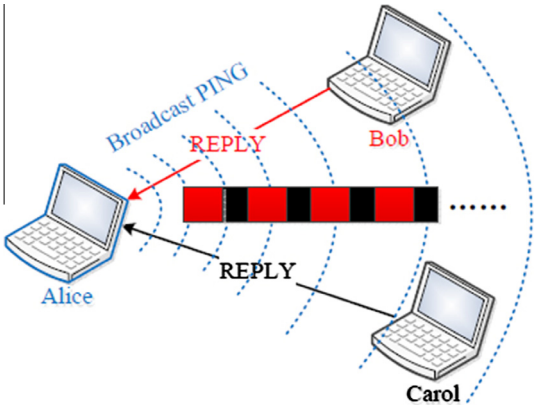


Fig. 7. Three-user probing scenario based on the shared channel approach.

5. Experimental results

First we show the advantage of legitimate users compared with the adversary user. We give the performance of the fuzzy controller under different situations, such as different velocities, different motion types, different sites, and different desired-KGRs. Then we compare the experimental results with the PID controller from our previous study [1,34].

5.1. Advantage of legitimate channel vs. eavesdropping channel

In shared key generation, the advantage of legitimate users should be ensured compared with an adversary user [19]. In other words, the legitimate users need to share more mutual information than the adversary user. Otherwise the process needs an extra process called *advantage distillation* or *preference distillation*.

We designed a three-user experiment in which Eve stayed close to Bob, and Alice could move randomly. After a certain duration of channel probing, each user observed a RSS sequence to be quantized. The results in Table 2 demonstrate that the Alice–Bob pair channel has significant higher mutual information than the Alice–Eve and Bob–Eve pair channels, i.e. $I(Alice; Bob|Eve) > I(Alice(or\ Bob); Eve)$. Note that a similar conclusion can be obtained at other probing rates and among more users.

5.2. Performance comparison of fuzzy and PID controllers

The performance results using the fuzzy controller with different velocities, motion types, and sites are shown in Table 3, with desired-KGR at 75 bits/s. The corresponding performance results using the PID achieved in our previous study are also provided in parenthesis for comparison.

Different Velocities. In the motion experiment, Alice's movements are designed as straight line walking, jogging, and running, respectively. The KGR error represents the precision using controllers; large value of the error indicates system failure. As shown in Table 3, when the user moves faster, the KGR error becomes larger, from 0.09 to 0.37 bits/s and the oscillation frequency of KGR increases while the overshoot decreases. Furthermore, the KGR reaches the setpoint more quickly if the user moves faster.

The ITAE values become smaller when Alice moves fast. As one can see, in this case the fuzzy controller performs better. Fast motion of the user results in high probing efficiency.

Different Motion Types. Random motion has a large KGR mean error of 0.31 bits/s, as compared a the straight-line motion mean error of 0.09 bits/s. Random motion also results in smaller oscillation frequency and overshoot. However, the KGR reaches the setpoint at the same time in both cases. In general, as the ITAE becomes small, the fuzzy controller performs better when the user moves randomly. Random motion also produces much higher probing efficiency than linear motion.

Different Sites or Locations. We also conducted an experiment in indoor and outdoor locations, as shown in Fig. 6.

Table 2
Mutual information among users (mobile and static), Probing rate $f = 100$ Hz.

Mutual information	Mobile (bits/s)	Static (bits/s)
I_{ab}	65.0	52.3
I_{ae}	4.1	3.0
I_{be}	2.3	0.9

The mobile velocities were nearly same (about 0.3 m/s) and motion types were both random motion. The indoor scenario has a large KGR mean error of 0.20 bits/s, as compared with the outdoor scenario mean error of 0.09 bits/s. The indoor scenario also results in a small oscillation frequency and overshoot. However, the KGR in both cases reaches the set-point at the same time.

In general, as the ITAE value is small, the fuzzy controller for indoor case performs better. Moreover, the indoor scenario produces much higher probing efficiency than the outdoor does.

Different Desired-KGRs. The desired-KGRs from all experiments above were set at 75 bits/s. Table 4 gives the results under different desired-KGRs. This experiment was conducted only for an indoor location with random motion and speed at about 0.3 m/s. With the increase of the desired-KGR, the probing rate increases but the efficiency drops. When the desired-KGR is smaller than 25 bits/s, the efficiency is very high, even above 50%. In general, the faster the users want to generate a key, the lower efficiency the channel probing has.

Since the maximum randomness of the wireless channel is definite, it is unable to provide very large KGRs. Desired-KGR values larger than 210 bits/s cannot be achieved in our platform.

From the experimental results above, we find that all the KGR errors are smaller than 0.38 bits/s with the desired-KGR value of 75 bits/s. Even when the user moves very fast, it only goes up to 0.37 bits/s. This indicates that the fuzzy controller has higher precision. The ITAEs are all smaller than 67, meaning that the fuzzy controller has good control ability for all kinds of situations.

Comparison of FL and PID Controllers. In order to compare the fuzzy controller with PID controllers, we made our best effort to ensure that both controllers have the same conditions. The corresponding performances results using the PID were extracted from the previous study [34]. The corresponding data are provided in parentheses in Table 5.

The detailed comparison of the fuzzy controller with the PID controller is plotted in Fig. 8. As shown in Fig. 8a, the KGR error and the ITAE of the fuzzy controller are both smaller than that of the PID controller. The fuzzy controller improves the accuracy by 0.11 bits/s, which is 36.7%. It enhances the control ability (ITAE) by 8.5 units which is 14.0% on average. This means the fuzzy controller is able to control the probing process much better. The higher the probing accuracy is, the smaller the overshooting and the lower the oscillation number.

Fig. 8b shows that the fuzzy controller in the channel probing is more efficient than the PID controller. The fuzzy controller improves the probing efficiency by 0.046 bits/s, which is 13.7% on average. However, it is hard to tell the probing rate difference between these two controllers because the channel randomness is not influenced by the controller types but by the environmental changes. Fig. 9 also supports this observation. The fuzzy controller has higher probing efficiency than the PID controller at different desired-KGR values and improves the probing efficiency by 0.10 bits/s, which is about 63.5% on average. Again, the difference in the probing interval between two controllers is not clear.

5.3. Multi-user probing

The experimental results in the three-user case (Alice, root node; Bob and Carol member nodes) are presented. Table 5 lists the DWF values corresponding to the satisfaction of the KGR requirements when desired-KGR values of these pairs (75 and 10 bits/s for Bob and Carol, respectively) are given. The DWF values can be determined based on the DWF scale model given in Section 3.1. For simplicity, we say Bob instead of the Alice–Bob pair and Carol instead of the Alice–Carol pair. The left-hand-side of the semicolons in Table 5 stands for Bob's satisfaction answer and the right-hand-side refers to Carol's. If the user's answer is satisfied, we mark 'Y'; if the user's answer is not satisfied, we mark 'N' and give the actual KGR. After observing the results in Table 5, we can state:

- Each user's DWF affects the other's actual KGRs. If Bob upgrades his DWF value, both Bob and Carol's actual-KGR values increase.
- If one user's desired-KGR value is smaller than the other's, his KGR requirement will be satisfied much more easily. In this example, Carol's desired-KGR is

Table 3
Performance of fuzzy controller under different situation (PID controllers' results are also listed in parenthesis).

Situation	Different velocities (m/s)			Different motion types		Different sites	
Values	0.4	1.0	1.5	Straight Line	Random	Outdoor	Indoor
KGR Error	0.09 (0.15)	0.25 (0.38)	0.37 (0.59)	0.09 (0.15)	0.31 (0.42)	0.09 (0.15)	0.20 (0.33)
KGR Osc. Freq.	0.26 (0.29)	0.31 (0.34)	0.32 (0.30)	0.26 (0.29)	0.33 (0.42)	0.26 (0.29)	0.23 (0.26)
KGR Oversht. mean	8.45 (9.82)	6.32 (7.01)	5.81 (6.17)	8.45 (9.82)	5.98 (7.29)	8.45 (9.82)	5.57 (6.10)
Settling Time (loop)	4 (4)	3 (3)	3 (3)	4 (4)	4 (3)	4 (4)	4 (4)
ITAE	66.1 (72.5)	50.4 (61.3)	43.7 (55.2)	66.1 (72.5)	50.9 (58.6)	66.1 (72.5)	39.8 (49.7)
Probing Rate	101.1 (102.4)	88.3 (89.1)	78.5 (80.1)	104.1 (102.4)	90.9 (92.4)	104.1 (102.4)	84.1 (83.4)
Efficiency (LZ76)	0.347 (0.318)	0.369 (0.320)	0.377 (0.337)	0.347 (0.318)	0.442 (0.345)	0.347 (0.318)	0.410 (0.362)

Table 4
Performance of fuzzy controller under different desired-KGRs.

Desired KGR	KGR mean	Probing rate	Efficiency (LZ76)
25	24.8	39.1	0.572
50	50.1	117.4	0.448
75	74.8	165.2	0.398
100	101.2	211.8	0.325
150	149.1	263.5	0.282
200	202.5	294.0	0.175
300	207.0	298.6	0.170

much smaller than Bob's and her KGR requirement can be satisfied much easily, no matter what DWF value she or Bob has.

- Probing Rate and Efficiency. Since Alice broadcasts the probing frame to both Bob and Carol, she determines the broadcast probing rate. She compares the weighed-actual-KGR $\tilde{K}$ of both pairs (Alice–Bob and Alice–Carol) with the weighed desired-KGR $\tilde{k}$. She is also responsible to determine a new broadcast probing rate for next loop.

In Fig. 10, we compare the probing rate and efficiency for a multi-user cluster. Here, the x-axis is marked by the pair $(w_b; w_c)$, where w_b is Bob's DWF and w_c is Carol's. We can see that the DWF upgrades increase the broadcast probing rate steadily from 5.2 to 133 and enlarge the deviation. As a trade-off, the probing efficiency decreases from 0.62 bits/s to 0.27 bits/s.

Let us consider the scenario with three users with different desired-KGRs and different DWFs.

Both Users in Trivial Grade. In Fig. 11 we specify that both Bob's and Carol's DWFs are in trivial grades, i.e. $w_b = 0, w_c = 0$. No matter what the users' desired-KGRs are, the actual-KGRs of the users fluctuate randomly with a small deviation. In other words, their actual KGRs are independent of their desired-KGRs. This is because the weighed desired-KGR directly equals the compensate value α when $w = 0$. The mean that Bob's and Carol's actual KGRs are 4.96 and 4.99 bits/s with standard deviation 0.048 bits/s and 0.094 bits/s, respectively. It is reasonable to say that if the number of users increases beyond three, the actual KGRs of all users are around α and the values are independent of each other. This scenario can be applied to the case when network and computational resources are temporally inadequate. Setting α at an appropriate value is very important to ensure that a multi-user probing system can operate normally and rationally in trivial grading.

Only One User in Trivial Grade. In Fig. 12 we see the scenario with Bob's DWF in normal grade while Carol's is in trivial grade ($w_b = 1, w_c = 0$). No matter what Carol's desired-KGR is, Bob's actual-KGR is determined by Bob's desired-KGR and Carol's actual-KGR is proportional to Bob's desired-KGR, respectively.

However, when Bob's DWF value is small, the probing system only provides a basic level of KGR, but with high efficiency. This will result in dissatisfaction by users who expect large desired-KGRs. Bob's actual KGR deviates from the desired-KGR value by an average of 1.6% in the present experiment. Since Carol's requirement does not influence the probing, her actual KGR deviates from the desired-KGR value by an average of 67.9%.

Both Users in Normal Grade. In this case, both Bob and Carol set their DWFs in normal grade ($w_b = 1, w_c = 1$). Even though Bob's desired-KGR is fixed, his actual KGR is determined by his and Carol's desired-KGRs. If Carol has more KGR requirements, Bob's actual KGR value is high. In general, the actual BGRs of Bob and Carol are not identical. For example, if Bob's and Carol's desired-KGRs are 75 bits/s and 25 bits/s respectively, their actual KGRs are 55.7 bits/s and 55.1 bits/s, respectively.

This analysis of experimental results indicates that one user's actual KGR is determined by all users' desired-KGRs and DWFs. Similar to the two-user scenario, both movement and environmental dynamics affect the final probing efficiency.

Parameters γ and α. In the three-user scenario above, we set the parameters γ and α in Eq. (2) as 0.5 and 5, respectively. What happens if we change γ and α? The compensation parameter α makes the probing system work with a reasonable probing rate when all users' DWFs are set in trivial grade because "user in trivial grade" means he/she does not care about satisfying his/her KGR requirement. In this case the system does not need to probe fast, which can save network resources. Therefore, it is better to set the compensation parameter α low, like the 5 bits/s in our experiments.

The compromising coefficient γ is very important. The selection range is from 0 to 1, and a large γ value would increase the desired-KGR weighting. In that case, users' KGR requirements would be satisfied much more easily, but the price to be paid is a decrease in probing efficiency. By contrast, a small γ would probably disappoint users who have large DWFs.

In practice, choosing appropriate values for γ often depends on the network and power resources. For example, it is much better to choose a small γ in ad hoc networks that are made up of power-limited and hand-held devices,

Table 5
Desired-key-generation-rate weighting factor for Alice–Bob and Alice–Carol cannel pairs.

Bob and Carol satisfied?		WFD of Carol with desired-KGR (10 bits/s)			
		Trivial	Normal	Anxious	Essential
WFD of Bob with desired-KGR (75 bits/s)	Trivial	N(5.2); N(5.5)	N(10.3);Y	N(15.8);Y	N(19.8;Y)
	Normal	N(42.2);Y	N(47.1);Y	N(53.0);Y	N(58.2);Y
	Anxious	Y;Y	Y;Y	Y;Y	Y;Y
	Essential	Y;Y	Y;Y	Y;Y	Y;Y

Question: Are they satisfied? (Y: yes; N: no)

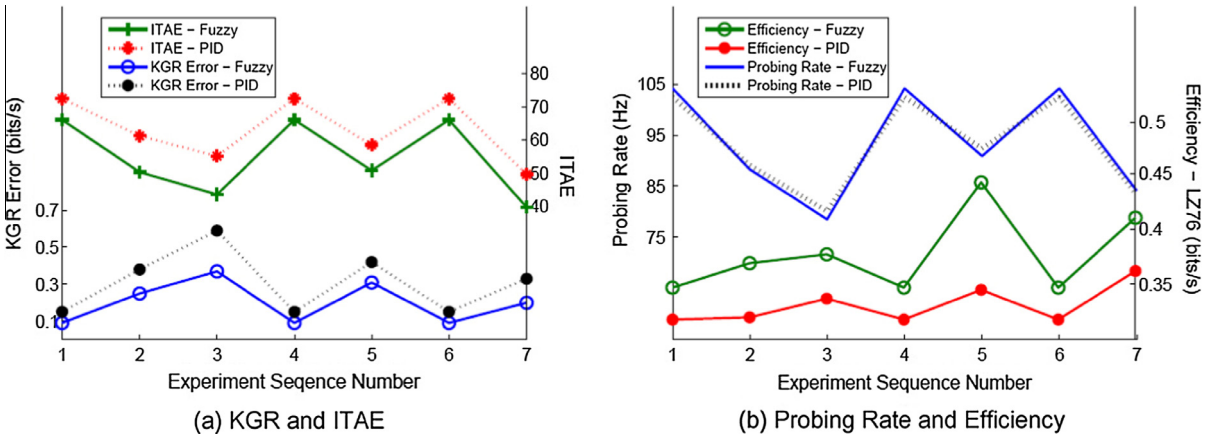


Fig. 8. Two process controllers in different scenarios. The *x*-axis stands for the series of experiments shown in Table 3.

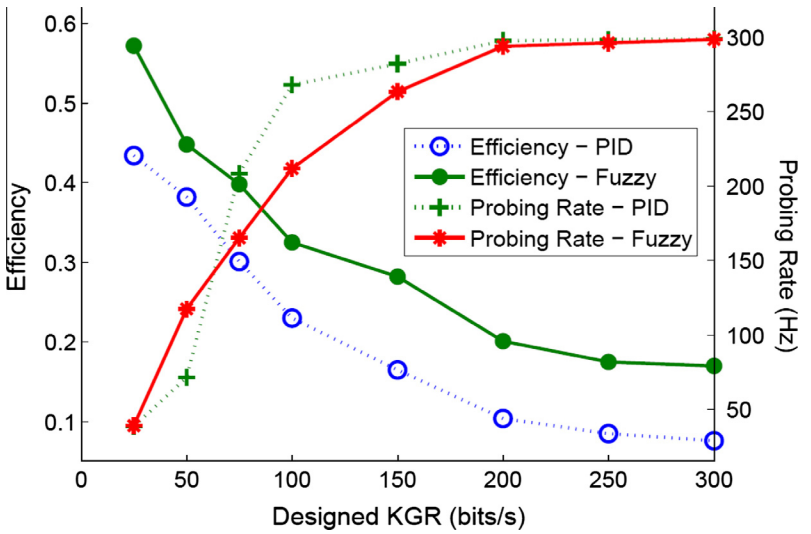


Fig. 9. Probing rate and efficiency under different desired-KGRs.

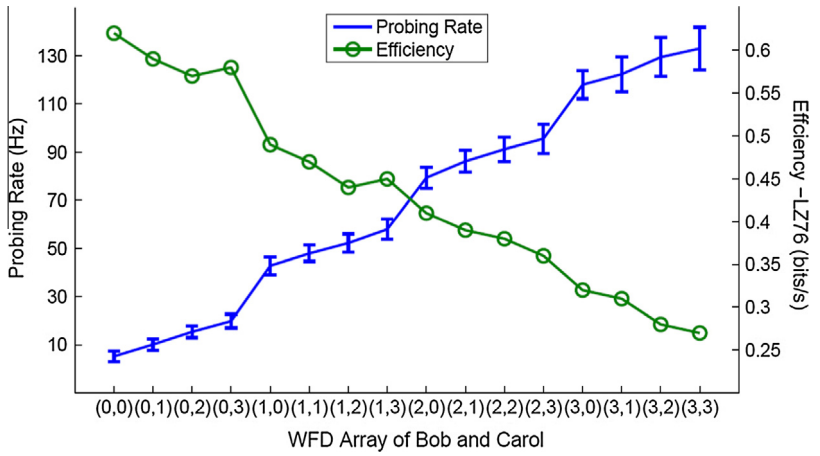


Fig. 10. Probing rate and efficiency of multi-users.

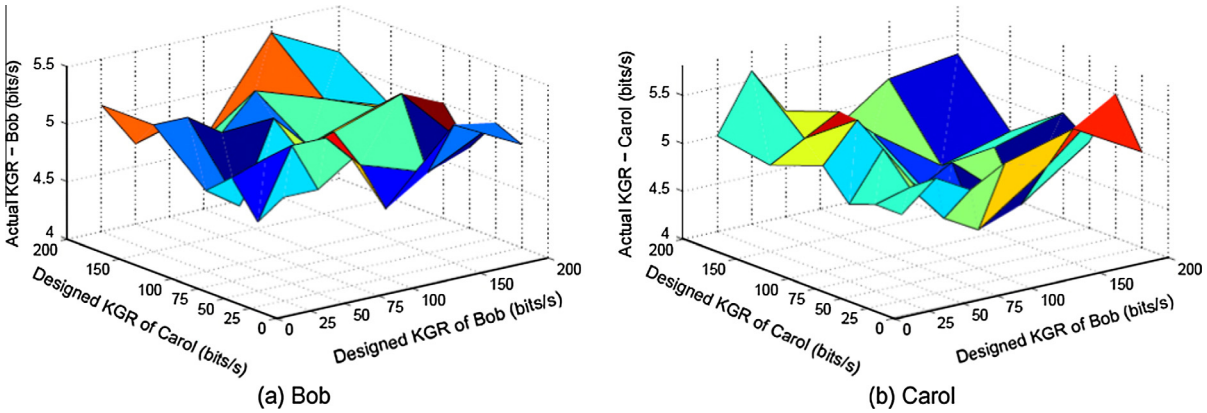


Fig. 11. Actual KGR, DWF (Bob: trial; Carol: trivial).

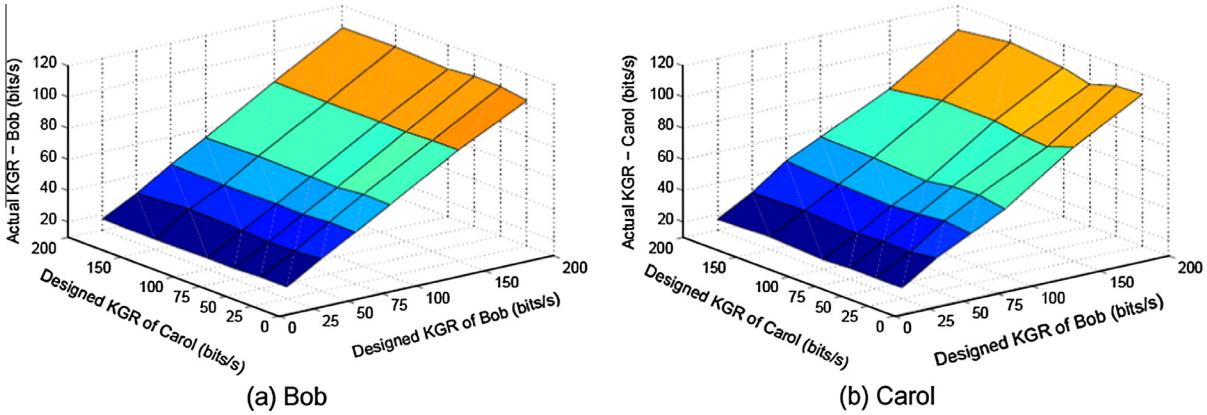


Fig. 12. Actual KGR and DWF (Bob: normal; Carol: trivial).

or in wireless sensor networks. By contrast, for cell communication networks with adequate energy resources and high bandwidth facility, it seems better to choose a large value.

6. Conclusion and discussion

In order to provide a feasible mechanism for multi-user key generation in wireless networks and to improve the performance of wireless channel probing systems, we developed a scheme to weight users' requirement satisfaction for multi-user key generation. Strategically, we applied a fuzzy controller to reduce the error between a user's desired-KGR and actual KGR. The experimental results show that the probing scheme with the fuzzy controller enables us to efficiently tune the probing rate under different situations, effectively produce smaller overshoots, and reduce oscillations as compared with that the probing scheme using a PID controller.

The most important contribution of this paper for probing systems is its application in a multi-user scenario for the first time, to the best of our knowledge. We introduced a broadcast probing approach and the Desired-KGR Weighting Factor (DWF). A series of three-user experiments demonstrated that the probing system enables us to make a trade-off between probing efficiency and the KGR requirement.

The fuzzy controller stabilizes the KGR at desired values with an error less than 0.38 bits/s and probing efficiency exceeding 0.44. Employing a fuzzy controller in the proposed scheme can improve the control accuracy by 0.11 bits/s or 36.7%, enhance the control capability by 8.5 units or 14.0% ITAE of the PID controller, and increase the entropy rate by 0.046 bits/s or 13.7% on average compared with the results using PID controller.

The multi-user channel probing method is the groundwork for group key generation in wireless networks. Although the present study is our primary research work, it provides a significant insight into further investigation in group key generation systems, such as agreements and policies among the nodes, security settling, and key refreshment.

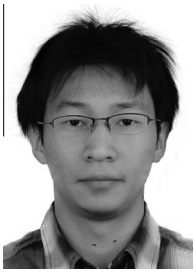
Acknowledgments

We appreciate the mentoring efforts of Prof. Prasant Mohapatra (Department of Computer Science, University of California at Davis). We also appreciate Prof. Kai Zeng (Department of Computer and Information Science, University of Michigan at Dearborn), Dr. Bin Xin (Decision and Cognitive Research Centre, Manchester Business School, the University of Manchester), and Dr. Steve Cunningham and Judy Brown (Brown and Cunningham Associates) to improve the quality of this paper.

References

[1] Y. Wei, K. Zeng, P. Mohapatra, Adaptive wireless channel probing for shared key generation, in: Proceedings IEEE INFOCOM, 2011, pp. 2165–2173.
[2] S. Jana, S.N. Premnath, M. Clark, S.K. Kasera, N. Patwari, 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, ACM, New York, NY, USA, 2009, pp. 321–332.
[3] S. Mathur, W. Trappe, N. Mandayam, C. Ye, 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, ACM, New York, NY, USA, 2008, pp. 128–139.
[4] K. Zeng, D. Wu, C. A, P. Mohapatra, Exploiting multiple-antenna diversity for shared secret key generation in wireless networks, in: Proceedings IEEE INFOCOM, 2010, pp. 1–9.
[5] N. Patwari, J. Croft, S. Jana, S. Kasera, High-rate uncorrelated bit extraction for shared secret key generation from channel measurements, IEEE Transactions on Mobile Computing 9 (1) (2010) 17–30.
[6] T. Aono, K. Higuchi, M. Taromaru, T. Ohira, H. Sasaoka, Wireless secret key generation exploiting the reactance-domain scalar response of multipath fading channels: RSSI interleaving scheme, in: Proceedings of The European Conference on Wireless Technology, 2005, pp. 173–176.
[7] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, 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.
[8] R. Wilson, D. Tse, R. Scholtz, Channel identification: Secret sharing using reciprocity in ultrawideband channels, in: Proceedings of IEEE International Conference on Ultra-Wideband, 2007, pp. 270–275.
[9] U.M. Maurer, Secret key agreement by public discussion from common information, IEEE Transactions on Information Theory 39 (3) (1993) 733–742.
[10] G. Brassard, 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.
[11] C. Cachin, U.M. Maurer, Linking information reconciliation and privacy amplification, Journal of Cryptology 10 (1997) 97–110.
[12] H. Koorapaty, A. Hassan, S. Chennakeshu, Secure information transmission for mobile radio, IEEE Communications Letters 4 (2) (2000) 52–55.
[13] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, N. Mandayam, Information-theoretically secret key generation for fading wireless channels, IEEE Transactions on Information Forensics and Security 5 (2) (2010) 240–254.
[14] R. Ahlswede, I. Csiszar, Common randomness in information theory and cryptography, part I. Secret sharing, IEEE Transactions on Information Theory 39 (4) (1993) 1121–1132.
[15] C. Chen, M. Jensen, Secret key establishment using temporally and spatially correlated wireless channel coefficients, IEEE Transactions on Mobile Computing 10 (2) (2011) 205–215.
[16] Q. Wang, H. Su, K. Ren, K. Kim, Fast and scalable secret key generation exploiting channel phase randomness in wireless networks, in: Proceedings IEEE INFOCOM, 2011, pp. 1422–1430.
[17] C. Ye, A. Reznik, Group secret key generation algorithms, in: IEEE International Symposium on Information Theory, ISIT, 2007, pp. 2596–2600.
[18] S. Nitinawarat, C. Ye, A. Barg, P. Narayan, A. Reznik, Secret key generation for a pairwise independent network model, IEEE Transactions on Information Theory 56 (12) (2010) 6482–6489.
[19] C.H. Bennett, G. Brassard, C. Crkpeau, U.M. Maurer, Generalized privacy amplification, IEEE Transactions on Information Theory 41 (6) (1995) 1915–1923.
[20] A.D. Orebaugh, Gilbert. Ramirez, Ethereal Packet Sniffing, Syngress (2004). February.
[21] L. Chappell, Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide, second ed., Podbook Press, 2012.
[22] A. Lempel, J. Ziv, On the complexity of finite sequences, IEEE Transactions on Information Theory 22 (1) (1976) 75–81.
[23] J. Thomas, Elements of Information Theory, John Wiley and Sons, 1991.
[24] S.P. Strong, R. Koberle, de Ruyter van Steveninck, W. Bialek, Entropy and information in neural spike trains, Physical Review Letters 80 (1998).
[25] J.M. Amigo, J. Szczepanski, Elek Wajnryb, M.V. Sanchez-Vives, Estimating the entropy rate of spike trains via Lempel–Ziv complexity, Neural Computation 16 (2004) 717–736.
[26] K.M. Passino, S. Yurkovich, Fuzzy Control, Addison-Wesley, Menlo Park, CA, 1998.
[27] Jan Jantzen, Foundations of Fuzzy Control, Wiley, 2007.
[28] G. Gerla, Fuzzy Logic Programming and Fuzzy Control, vol. 79, Studia Logica, 2005.
[29] W.A. Wali, J.D. Cullen, K.H. Hassan, A. Mason, A.I. Al-Shamma'a, Comparison between PID and fuzzy logic controllers for advance microwave biodiesel reactor, in: Proceeding of IEEE Symposium of Computers and Informatics, 2001, pp. 346–351.
[30] J. Pereira, J.B. Bowles, Comparison of PID and fuzzy controls of a model of car, in: Proceedings of the Third IEEE Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence, June 1994, pp. 26-29.
[31] M. Santos, S. Dormido, J. M. de la Cruz, Fuzzy-PID vs. fuzzy-PI controllers, in: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, vol. 3, 1996, pp. 1598–1604.
[32] E. Natsheh1, K.A. Buragga, Comparison between conventional and fuzzy logic PID controllers for controlling DC motors, IJCSI International Journal of Computer Science Issues 7 (5) (2010).
[33] R. Merkle, Secrecy, Authentication, and Public Key Systems, Ph.D. Thesis, Stanford University, 1979.
[34] Y. Wei, Kai Zeng, P. Mohapatra, Adaptive wireless channel probing for shared key generation based on PID controller, IEEE Transactions on Mobile Computing (4) (2011) 207–219.

Dr. Lihua Dou is currently a professor in the Department of Automation at Beijing Institution of Technology, Beijing, China. She received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from Beijing Institute of Technology in 1979, 1987, and 2001, respectively. Her research interests include multi-objective optimization and decision, pattern recognition, image processing and security in wireless networks.



Yunchuan Wei is currently an engineer in Research and Development Center of China Academy of Launch Vehicle Technology, Beijing, China. He received his B.E and Ph. D. degrees from Beijing Institute of Technology in 2006 and 2012, respectively. He was a visiting Ph.D. candidate under the supervision of Prasant Mohapatra in the Department of Computer Science at University of California, Davis, supported by China Scholar Council from 2009 to 2010. His research interests include physical layer security, ad hoc networks and automatic control.

Dr. Jun Ni is currently an associate professor in the departments of radiology, computer science, mechanical engineering, and biomedical engineering at the University of Iowa. He also serves as a chair professor in the School of Information Science and Technology of Shanghai Sanda University and as a visiting professor at Harbin Engineering University and Nanjing University of Science and Technology. He received his B.S. from Harbin Engineering University, M.S. from Shanghai Jiaotong University, and Ph.D. from the University of Iowa, in 1982, 1984, and 1991, respectively. He did his postdoctoral work at the University of Iowa and Purdue University, respectively from 1992-1994. His research interests include thermo-fluid science, materials processing, high performance computing, information technology, multi-scale computational modeling and simulation, medical data/image processing, medical and healthcare informatics, wireless networks, and systems engineering.