Secret Key Extraction using Bluetooth Wireless Signal Strength Measurements

Prarthana L. Gowda*, Sriram N. Premnath†, Sneha K. Kasera*, Neal Patwari‡, Robert Ricci*
*School of Computing, University of Utah, {gowda, kasera, ricci}@cs.utah.edu
†School of Electrical & Computer Engineering, Cornell University, sriram.np@cornell.edu
‡Department of Electrical & Computer Engineering, University of Utah, npatwari@ece.utah.edu

***Abstract*—Bluetooth has found widespread adoption in phones, wireless headsets, stethoscopes, glucose monitors, and oximeters for communication of, at times, very critical information. However, the link keys and encryption keys in Bluetooth are ultimately generated from a short 4 digit PIN, which can be cracked off-line. We develop an alternative for secure communication between Bluetooth devices using the symmetric wireless channel characteristics. Existing approaches to secret key extraction primarily use measurements from a *fixed, single* channel (e.g., a 20 MHz WiFi channel); however in the presence of heavy WiFi traffic, the packet exchange rate in such approaches can reduce as much as 200×. We build and evaluate a new method, which is robust to heavy WiFi traffic, using a *very wide bandwidth* ($B \gg 20$ MHz) in conjunction with *random frequency hopping*. We implement our secret key extraction on two Google Nexus One smartphones and conduct numerous experiments in indoor-hallway and outdoor settings. Using extensive real-world measurements, we show that *outdoor settings are best* suited for secret key extraction using Bluetooth. We also show that even in the absence of heavy WiFi traffic, the performance of secret key generation using Bluetooth is comparable to that of WiFi while using much *lower transmit power*.**

I. Introduction

There has been a tremendous growth in the proliferation of pervasive computing devices such as smartphones, tablets, and medical devices. In general, these devices are bundled with an array of wireless technologies, mainly, cellular, WiFi, and Bluetooth. Bluetooth, in particular, has found widespread adoption in phones, wireless headsets, stethoscopes, glucose monitors, and oximeters for communication of, at times, very critical information [1]. However, serious security vulnerabilities have been discovered for Bluetooth [2]. For instance, the link keys and encryption keys in Bluetooth are ultimately generated from a short 4 digit PIN, which can be cracked off-line [2], [3], [4]. In this paper, we develop an alternative secure communication method for Bluetooth devices that is capable of producing arbitrarily long secret keys, which when used as one-time pad, can provide security against adversaries with unlimited computational power.

We propose to exploit the characteristics of the *wireless channel* between two Bluetooth nodes, Alice and Bob, that varies with time and space for generating secret keys. The wireless channel between any two nodes is *symmetric/reciprocal* since the multi-path properties of the channel which include gains, phase shifts and delays are identical on both directions of the link. These wireless channel characteristics vary with time due to changes in the environment such as the movement of objects or people. Moreover, the multi-path characteristics are location-specific – an eavesdropper, Eve, who is separated by a few wavelengths from Alice and Bob, cannot measure the same multi-path characteristics [5], [6] (Figure 1). Thus, there is an *inherent secret* shared between any two wireless nodes that can communicate with each other directly, and we take advantage of this fact to establish a secret key between them. Secret key extraction from time and space varying wireless channels is not new. There exists a vast amount of work on this topic; however, there is no existing work on secret key extraction using Bluetooth. *The wide-spread use of Bluetooth and its unique strengths and limitations (described later in this paper) make it an interesting target for exploration for secret key extraction. This exploration is the focus of our paper.*

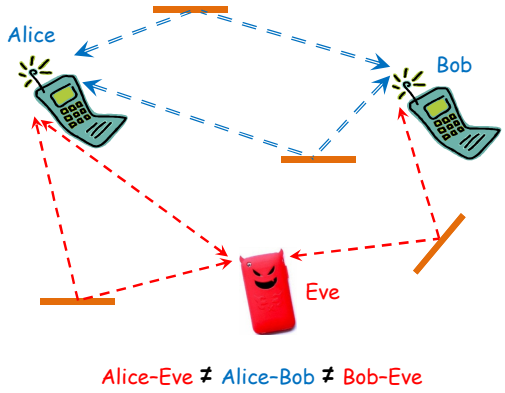

Fig. 1. Symmetric wireless channel between Bluetooth devices Alice and Bob.

Existing approaches (e.g., [6]) to secret key extraction primarily use measurements from a *fixed, single* channel (e.g., a 20 MHz WiFi channel as in [6]) for secret key establishment. Measuring the channel involves the exchange of packets between Alice and Bob. However, as we demonstrate later in this paper, in the case of WiFi, the packet exchange rate can reduce by as much as 200× in the presence of heavy traffic. Motivated by this observation, we build and evaluate a

978-1-4799-4657-0/14/$31.00

new method that we call *robust secret key extraction (RSKE)* using Bluetooth that uses a *very wide bandwidth* ($B \gg 20$ MHz) in conjunction with random *frequency hopping* over a set of narrow channels within B to avoid those frequencies that are heavily used. We experimentally show that with our approach using Bluetooth, we can exchange packets and collect measurements for secret key extraction *even under heavy WiFi traffic*, *an order of magnitude* faster in comparison to using WiFi.

Our RSKE system comprises of different stages including the more traditional sampling, quantization [6], information reconciliation [7], and privacy amplification [8], and a more recently proposed interpolation [9] stage. In the sampling stage, Alice and Bob exchange a series of packets to collect the received signal strength (RSS) measurements. Then, Alice and Bob individually estimate their half-duplex measurements at common time instants using the interpolation stage to reduce asymmetry in their measurements. Next, they quantize their measurements to obtain an initial bit sequence [6]. Then, they reconcile potential mismatches in their initial bit sequence using the information reconciliation stage [7] and finally, they use the privacy amplification [8] stage to obtain a high entropy secret bit sequence.

We implement our secret key extraction on two Google Nexus One smartphones and using extensive real-world measurements from these phones, we show that (i) inclusion of an interpolation stage in secret key extraction, significantly reduces the mismatch in the quantized bits of Alice and Bob, (ii) outdoor settings achieve significantly higher secret key generation rate in comparison to indoor-hallway settings, (iii) secret bit sequences that we obtain using Bluetooth RSS values pass eight different randomness tests of the NIST test suite that we conduct, (iv) *even in the absence of heavy WiFi traffic*, RSKE using Bluetooth can achieve secret key generation performance that is comparable to using WiFi even with the use of much lower transmit power.

II. Secret Key Generation Methodology

In the first step of the secret key generation process, namely, sampling, Alice and Bob collect a time series of RSS measurements by exchanging packets. In an earlier work [6], Premnath et al. have used a three stage process to convert a set of RSS measurements into a secret bit sequence. These three stages are quantization, information reconciliation and privacy amplification. For secret key extraction using Bluetooth RSS measurements, we also consider an additional stage called interpolation from Patwari et al.'s work [9]. Figure 2 shows the various stages in the secret key extraction process. We discuss each of these steps in detail as follows.

A. Quantization of RSS samples

Quantization is the process of converting a sequence of RSS measurements into a sequence of bits. Alice and Bob perform the following steps in generating an initial bit sequence: (i) define an upper threshold q^+ and a lower threshold q^-, where $q^+ = \mu + \alpha \times \sigma$ and $q^- = \mu - \alpha \times \sigma$; here μ and σ represent the mean and standard deviation over a window of RSS measurements, with $\alpha \geq 0$. (ii) RSS values less than the lower threshold are encoded as zero; RSS values greater than the upper threshold are encoded as bit one and all RSS values that lie between the upper and the lower threshold are discarded. (iii) maintain a list of indices of discarded RSS values and exchange it with each other so that they exclude all such indices from further consideration for secret key extraction.

In the quantization of Bluetooth RSS values, we reduce the effects of shadow fading, which are caused due to obstructions from large objects in the environment. This ensures that the bit sequence obtained after quantization is mostly from the hard-to-predict effects of small scale fading. Note that small scale fading is caused because of relative motion between the radios and different objects in the environment. Premnath et al. [10] have shown that effects of shadow fading can be effectively filtered out if the window size, over which the quantization thresholds are computed, is chosen according to the sampling rate and the speed of nodes. *We incorporate this technique to calculate the right window size for the quantization of Bluetooth RSS values.*

B. Information reconciliation of quantized bits

The quantized bits sequences of Alice and Bob may have minor differences due to noise, interference, hardware manufacturing variations, half-duplex nature of communication, etc. To reconcile any differences in the bit sequences, we use the Cascade protocol [7]. In Cascade, Alice and Bob exchange parity information to locate and correct any bit mismatches iteratively until there is a high probability of success.

C. Privacy amplification of reconciled bit stream

When the channel is sampled faster than the rate at which it changes, there may be short term correlation in the subsequent bits of Alice and Bob. Moreover, a fraction of information, in the form of parity bits, is leaked into the insecure public channel during information reconciliation. Privacy amplification [8] addresses both these problems by applying universal hash function to generate shorter bit sequence with high entropy, where the output length of the hash function is chosen based on the entropy of the bit stream and also on the amount of information leaked during information reconciliation.

D. Interpolation of RSS samples

In order to deal with the asymmetry in RSS values at Alice and Bob that is caused due to the half-duplex nature of the transceivers, Patwari et al. [9] have used a pre-quantization interpolation step. Interpolation addresses the asymmetry by estimating the measurements of Alice and Bob at common time instants. We use the interpolation stage, in addition to the information reconciliation stage, to minimize the bit mismatch probability. We show in Section VII that interpolation is essential for secret key extraction using Bluetooth since it can be drastically reduce the percentage of mismatch in the quantized bits of Alice and Bob.

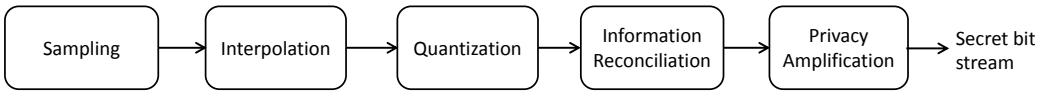

Fig. 2. RSS-based secret key extraction process

Let T_R denote the time delay between two subsequent measurements of Alice or Bob, and $\tau_a(i)$ and $\tau_b(i)$ denote the time instants at which Alice and Bob record their i^{th} measurements respectively. The fractional sampling offset, μ is calculated as

$$\mu = \frac{1}{2}\left[\frac{\tau_b(i) - \tau_a(i)}{T_R}\right]; \text{where } \tau_a(i) < \tau_b(i)$$

If Alice and Bob estimate their measurements at a delay of $(1+\mu)T_R$ and $(1-\mu)T_R$ respectively, then we would have simultaneous estimates for their measurements. In our work, we use this interpolation technique as a pre-quantization stage to reduce the probability of bit mismatches. The complete process of Bluetooth RSS-based secret key extraction that we use in this work is shown in Figure 2.

III. Adversary Model

We assume that the adversary, Eve, can listen to all conversations between the genuine parties, Alice and Bob. The adversary can also measure the channel between herself and Alice and Bob at the same time as them. However, Eve cannot be positioned very close to either Alice or Bob. Specifically, Eve should be several wavelengths away from both of them so that she measures an uncorrelated channel [5], [6]. The algorithms for key extraction along with the parameters used are public. We assume that the data integrity is protected, i.e., Eve is not interested in manipulating the messages between Alice and Bob. Eve is capable of affecting the wireless channel between Alice and Bob by moving objects randomly in the environment. However, she cannot control the movements to an extent as to significantly increase the coherence time of the channel. Bluetooth uses a standardized algorithm [11] to hop between different channels in the 2.4 GHz band, where the hopping sequence is common to all nodes in the Bluetooth piconet. We assume that Eve does not belong to the piconet and hence does not know the hopping sequence. However, she can jam the WiFi channels in the same 2.4 GHz band. Our scheme does not authenticate Alice or Bob, and is not immune to active person-in-the-middle attacks. The well-known Diffie-Hellman secret key establishment scheme has found wide-spread use in network security protocols and standards even without an authentication mechanism. We believe that our scheme will provide a strong alternative to the Diffie-Hellman scheme for establishing session keys in wireless networks.

IV. Robustness of Bluetooth to Heavy Traffic

Sampling, the first step in the secret key extraction process, involves the exchange of probe packets between Alice and Bob to collect the RSS values. In this section, we experimentally demonstrate that in the presence of heavy traffic, the sampling period could become excessively large for WiFi, while Bluetooth is robust to heavy traffic.

Consider a network with three nodes, A, B and C. Assume that nodes A and B are exchanging a high rate of data, for instance, streaming videos or exchanging large data files between them. Now, node C wants to establish a secret key with one of these nodes, say node A, using the characteristics of the wireless channel between itself and node A. If nodes A and C choose to use the existing WiFi channel, (i) nodes A and C will be able to sample the channel only at a very low rate for collecting RSS measurements, and (ii) it will slow-down the existing data transfer that is taking place between nodes A and B due to the new exchange of probe packets between nodes A and C for collecting RSS measurements.

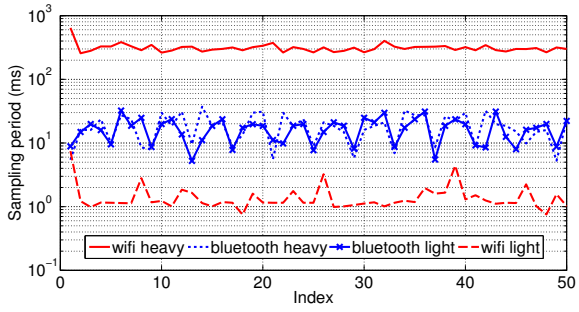

Fig. 3. Sampling period for WiFi and Bluetooth under heavy and light WiFi traffic.

We create an 802.11 ad hoc network with 3 nodes A, B and C for our simple experiment. Nodes A and B, which are about 4 ft to 7 ft from node C in an indoor setting, continuously exchange large UDP packets as fast as they can. In our experiment, nodes A and B were able to exchange about 9.5 Mbps of application layer data in each direction. Figure 3 shows the achievable sampling period over 50 consecutive successful packet exchanges.

Observation 1: While the WiFi sampling period is a mere 1.5 ms on average in the absence of UDP traffic, it increases more than 200× to about 314 ms on average when there is heavy UDP traffic! Note that this sampling period far exceeds the coherence time interval of indoor environments with typical mobility. As a result, there will be a great degree of asymmetry in the measurements (and consequently the quantized bits) of nodes A and C.

Observation 2: If nodes A and C exchange probe packets over the Bluetooth interface, on the other hand, the sampling period is about 18 ms, on average, with (or without) the exchange of WiFi UDP traffic between nodes A and B. Thus,

in the presence of heavy WiFi traffic, if we choose to use Bluetooth, we can collect a given number of RSS measurements at least *an order of magnitude* faster in comparison to WiFi there by significantly speeding up the secret key extraction process. Bluetooth achieves lower sampling period and remains robust to heavy WiFi traffic since it uses a *very wide bandwidth* ($B \gg 20$ MHz) in conjunction with random *frequency hopping* over a set of narrow channels within B to avoid those frequencies that are heavily used.

To summarize, Bluetooth is robust to heavy traffic; it is capable of achieving desirably low sampling period; and hence it is very promising for secret key extraction. Motivated by these results, we describe our robust secret key extraction approach in the next section.

V. Robust Secret Key Extraction Method

Existing approaches (e.g., [6]) primarily use measurements from a *fixed, single* channel (e.g., a 20 MHz WiFi channel as in [6]) for secret key extraction. However, these existing approaches perform poorly in the presence of heavy WiFi traffic – in fact, as we experimentally demonstrate in Section IV, the WiFi channel sampling rate can reduce by as much as two orders of magnitude. To address this problem, we develop a new mechanism, Robust Secret Key Extraction (RSKE), that uses a wider bandwidth B (with $B \gg 20$ MHz) and randomly hops over a set of narrow channels within B that specifically avoids those frequencies that are heavily used.

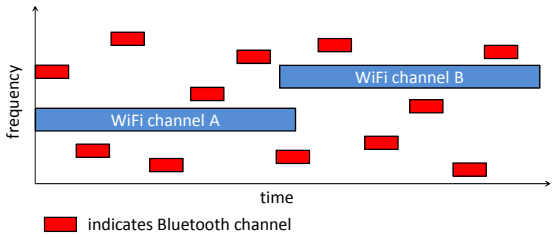

Fig. 4. Adaptive frequency hopping excludes frequencies that are heavily used.

We elucidate our RSKE approach using the frequency hopping spread spectrum (FHSS) mechanism of Bluetooth, in which the signals at the physical layer are transmitted by rapidly switching or hopping frequencies over a very wide bandwidth [12]. Bluetooth divides the 2.4 GHz band into 79 channels and hops at the rate of 800 times per second in a pseudo-random fashion. When some arbitrary wireless device in the vicinity operates in the 2.4 GHz frequency band, Bluetooth devices identify the channels used by the interfering device and remap the hopping sequence to exclude them. To achieve this exclusion of channels with interference, Bluetooth devices periodically estimate the channel usage and accordingly adapt their frequency hopping pattern. This technique is referred to as Adaptive Frequency Hopping (AFH) [13]. Figure 4 illustrates the operation of AFH mechanism in the presence of WiFi traffic on two different channels A and B. Note here that AFH specifically avoids the interfering WiFi channels.

Each Bluetooth device maintains an *AFH_channel_map*, which indicates the channels that are *occupied* by other interfering devices. The *AFH_channel_map* contains 79 one-bit fields, where each bit indicates the status of the corresponding channel. A bit 1 in the n^{th} position indicates that the n^{th} channel is used and similarly, bit 0 indicates a channel that is not used by other devices. Bluetooth can operate successfully when the hopping sequence includes a minimum of 20 channels out of a total of 79 available channels [11].

Bluetooth uses a hop sequence generator that takes in input parameters including *AFH_channel_map*, Bluetooth device address and clock of the *master*, etc., performs operations such as addition, XOR, permutation, etc. on these input parameters to output a pseudo-random sequence of channel indices [12], [11]. This sequence of indices is then mapped into actual frequencies in the 2.4 GHz band using the following relation: $f = 2402 + k$ MHz, $k = 0, ..., 78$, where k denotes the channel index. Note that the channels are spaced 1 MHz apart.

Obtaining symmetric RSS measurements: When a pair of Master and Slave Bluetooth devices, i.e., Alice and Bob, exchange a sequence of packets, each pair in this sequence – which includes a packet transmission from Alice to Bob and the corresponding response packet transmission from Bob to Alice – uses the same frequency. This ensures the *reciprocity/symmetry* of the measurements of Alice and Bob in a frequency selective fading channel.

We summarize the steps involved in the RSKE approach as follows.

1) Estimate channel usage over the frequencies in the interval $[F, (F + (N-1) \times 10^6)]$ and obtain the $channel\_map$; here F is the smallest carrier frequency, N is the number of channels and for Bluetooth, $F = 2402$ MHz and $N = 79$ channels.
2) Generate random hopping sequence S based on $channel\_map$; an arbitrary element in the sequence S at index i, $S(i) \in \{0, 1, 2, \ldots, (N-1)\}$ and $channel\_map(S(i)) = 0$.
3) Choose $f_i = F + S(i), \forall S(i) \in S$, where i is the packet sequence number.
4) Exchange packet pair (request from Alice and corresponding response from Bob) with sequence number i using frequency f_i.
5) Record RSS measurement (at both Alice and Bob) corresponding to sequence number i.
6) Jump to Step 1 periodically to assess whether a new hopping sequence has to be generated.
7) Once Alice and Bob collect the RSS measurements, they perform interpolation [9] and quantization [6]. Then, they reconcile potential mismatches in their initial bit sequence using information reconciliation [7] and finally they use privacy amplification [8] to obtain output secret bit sequence with high entropy.

While we have described the operation of RSKE in the context of Bluetooth, this approach is also applicable to other wireless technologies that use FHSS at the physical layer.

VI. IMPLEMENTATION

We use two Google Nexus One smartphones in our implementation which are equipped with BCM4329EKUBG Broadcom chips [14]. These support a Bluetooth Core Specification Version 2.1 with enhanced data rate technology. Both the phones run the Android 2.1 operating system.

In our implementation, we use the Bluetooth BlueZ protocol stack [15] for Android. Programming the BlueZ protocol stack is significantly simpler than using two other possible options – Android Native Development toolkit (NDK) and Android Software Development toolkit (SDK).

A. Methods for Sampling the Bluetooth Channel

1) Inquiry based RSS Method: The *inquiry*[1] *mode with RSS* is the only method to obtain *actual raw* RSS values. The Host controller Interface (HCI) of the BlueZ protocol stack [15] provides functions to configure a device in the *inquiry mode with RSS*, which allows one to obtain the RSS value for any inquiry response message received from other devices. While the inquiry based RSS method allows one to obtain *raw* RSS values, the following issues associated with this method make it unsuitable for secret key extraction.

First, due to the random back-off algorithm in the inquiry response state and due to the large time delays involved in the inquiry process, we can collect the RSS values only at a very low rate. For instance, in order to collect adequate response packets, an inquiry sub-state lasts for a duration of at least 10.4 seconds [11]. More importantly, the user cannot adjust the inquiry intervals to increase the sampling rate. Second, the inquiry message is a broadcast message and any device in the proximity can respond to this message. There is no way for the inquiring device to select one specific device that it needs response from with which it wants to establish the secret key.

Last, the devices must be in the discoverable mode in order to respond to an inquiry message. However, due to security concerns, current Bluetooth configuration for smartphones allows a device to be in discoverable mode for a maximum of 180 seconds only. A period of 180 seconds is too small to collect enough RSS measurements and extract keys of any reasonable size.

2) Connection based RSS Method: Using the alternative connection based RSS method, we can obtain an RSS measurement for every data packet received over a connection established between two Bluetooth devices at very high rate between 18 Hz to 24 Hz. However, this method does not return the raw RSS values. Instead, the returned RSS value indicates whether the actual measurement is above, below, or within the *Golden Receive Power Range (GRPR) [12].*

The GRPR, which is considered as the ideal received power range, is defined using a lower and an upper threshold. The lower threshold corresponds to a received power between -56 dBm and 6 dB above the actual sensitivity of the receiver. The upper threshold is 20 dB above the lower threshold level with an accuracy of ± 6 dB [11]. A positive or negative RSS (in dB) indicates a received power level above or below the GRPR respectively, while a zero implies that it is ideal (i.e., within GRPR).

Importantly, note that this method of collecting RSS values results in a loss of many useful measurements. This is because, unless the RSS values lie outside the ideal range, these cannot be measured and will appear as a zero. We observe from our experiments that, for small distances such as 2 feet in a indoor hallway setting, the number of zeroes could be as high as 90% of the total number of samples. Even in the presence of large number of zeros in certain settings, the sampling rate is higher using the connection-based RSS method. Thus, we use this method in our work.

B. L2CAP packet exchange protocol

We develop a simple protocol using L2CAP sockets for collecting the RSS values. In this protocol, Alice sends a request L2CAP packet to Bob, who measures the RSS value for the received packet and sends back a reply L2CAP packet for which Alice measures the RSS value. We use the *hci_read_rssi* function to read the RSS value for each received packet. Alice and Bob continue to exchange these L2CAP request/reply packets to obtain a time series of RSS values.

We choose to use L2CAP for packet exchange since it provides a reliable connection oriented link that handles packet losses through retransmission. Information frames (I-frame) are used to carry information payload and supervisory frames (S-frame) are used to acknowledge information frames and request retransmissions, if necessary [12].

We implement the interpolation and quantization of RSS values as in [6]. For information reconciliation and privacy amplification, we implement the Cascade protocol [7] and universal hash functions respectively as in [6].

VII. EXPERIMENTATION

We first describe our experimental setup in Section VII-A, and then present our results in Section VII-B.

A. Experimental Setup

We conduct several experiments under two different mobile environments[2] (indoor-hallways and outdoors) with varying distances between Alice and Bob. In each environment, we perform five *walk-experiments*, where the phones are carried at normal walking speed and are separated by an average distance of x feet, where $x \in \{2, 5, 10, 20, 30\}$. In all of our experiments, Alice and Bob use a maximum transmit power of $3dBm$[3].

[1] A Bluetooth device uses inquiry mode to discover other devices in proximity.

[2] Using real world measurements of WiFi channel, Jana et al. [5], have established that mobile settings are best suited for secret key extraction.

[3] We experimentally find that beyond reasonably small distances (≤ 2.75 ft), in our Google phones, Bluetooth switches to maximum transmit power.

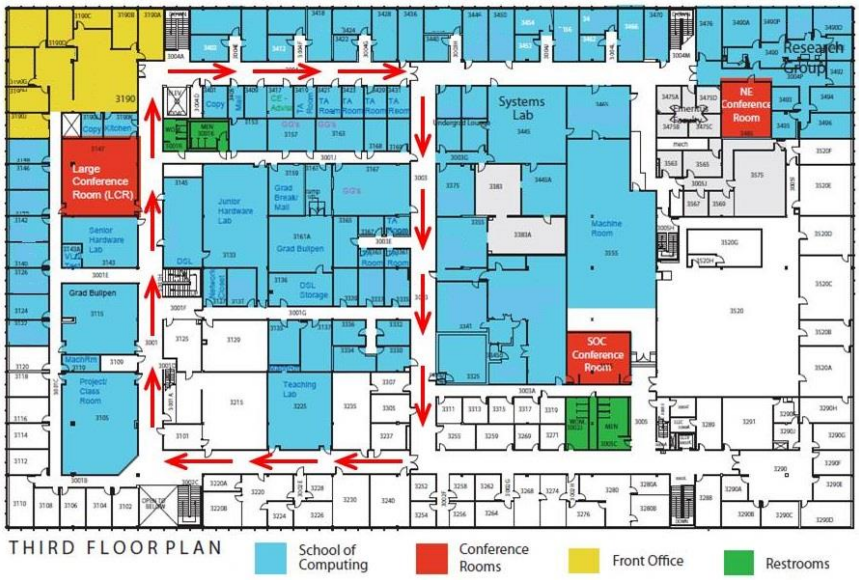

Fig. 5. Hallway experimental setup

Our indoor environment is a hallway on the third floor of our Engineering Building in our university campus. Figure 5 shows our experimental setup for the hallway environment and the path taken by Alice and Bob during these experiments.

We conduct a second set of experiments in an outdoor environment across varying terrain, with many trees, strolling people, pets and skate boarders. Alice and Bob are carried at normal walking speed from our Engineering Building to the Library in our university campus. Figure 6 shows the trajectory of Alice and Bob in the outdoor environment.

We collect 30000 RSS samples in each experiment. We find that the signal-to-noise-ratio (SNR) is higher in hallways in comparison to outdoor settings. While the sampling rate for hallways is around 24 Hz, variable packet loss rate under outdoors causes the sampling rate to change from 18 Hz to 24 Hz depending on the distance between Alice and Bob.

B. Results

We evaluate the performance of Bluetooth for secret key extraction using three metrics - (i) entropy, which captures the degree of randomness associated with each secret bit; (ii) bit mismatch rate, which is defined as the ratio of number of bits that do not match between Alice and Bob to the total number of bits extracted after the quantization step; and (iii) secret bit rate, which is defined as the average number of secret bits extracted per measurement; the secret bit rate is calculated in terms of the final output bits after accounting for bit losses during information reconciliation and privacy amplification stages.

1) Bit mismatch rate under hallway settings: Figure 7 shows the bit mismatch rate as a function of the quantization parameter, α for different distances under hallway settings; recall that α parameterizes the quantization thresholds $q = \mu \pm \alpha \times \sigma$; essentially, α controls the width of the censored region

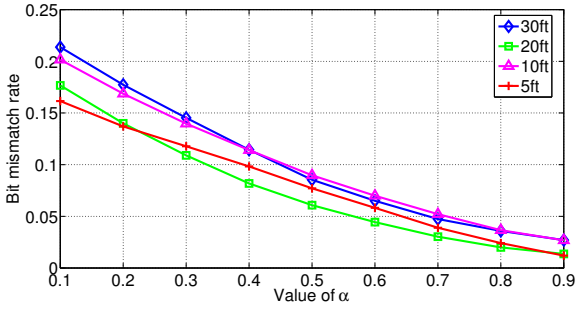

Fig. 7. Bit mismatch rate as a function of quantization parameter α for varying distances under hallway settings (without interpolation).

(i.e., the region between the upper and lower thresholds). When we consider the support of two correlated random variables, each representing a measurement from Alice and Bob, the area where these random variables (when quantized) disagree, becomes large as the area corresponding to the censored region is reduced. Thus the bit mismatch rate decreases with increase in the value of α.

We find that there is no clear relation between bit mismatch rates and distance under hallway settings because the SNR is high for all distances and the sampling rate does not change with distance, which is around 24 Hz for all distances. This is possibly because the hallway setting behaves like a wave guide, where the signal propagation is mostly confined within the hallway due to multiple reflections from the walls, ceiling and the floor [16]. As a result, there is minimal propagation loss with considerable variation in distance and consequently there is little difference in performance due to variation in distance under hallway settings.

Effect of interpolation: As we have discussed in Section II-D, the interpolation stage estimates the measurements

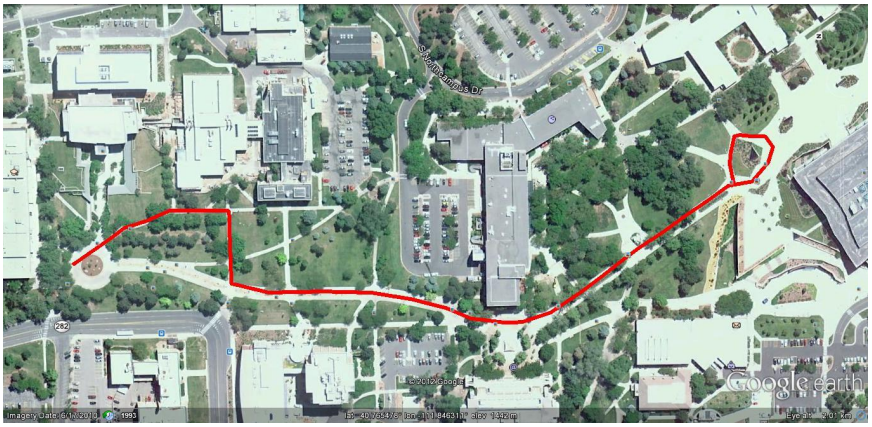

Fig. 6. Outdoor experimental setup (From Google Earth)

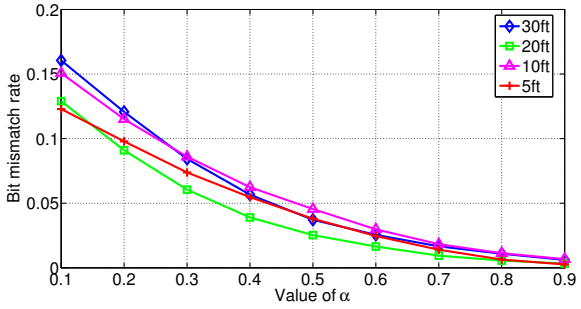

Fig. 8. Bit mismatch rate as a function of quantization parameter α for varying distances under hallway settings with the use of interpolation.

of Alice and Bob from a half-duplex channel at common time instants. We find that interpolation reduces the bit mismatch rate considerably as we have shown in Figure 8 (compare with Figure 7). However, the bit mismatch rate is still high enough under hallway settings to leak out a large fraction of information in the information reconciliation stage.

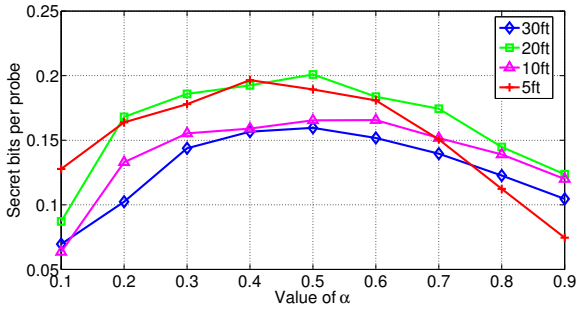

Fig. 9. Secret bit rate as a function of quantization parameter α for various distances under hallway settings with the use of interpolation.

2) Secret bit rate under hallway settings: Figure 9 shows the secret bit rate as a function of the quantization parameter, α for different distances under hallway settings, when interpolation is applied to the measurements of Alice and Bob. For each distance, note that above a particular α value, the secret bit rate decreases as a greater fraction of measurements is censored. However, the secret bit rate also decreases below that particular α value due to higher bit mismatch rate (with low α values) and the consequent increase in information leakage in the information reconciliation stage.

We show the secret bit rate results with the use of interpolation only since it increases the peak secret bit rate for hallway settings by up to 53.8%. Again there is no clear relationship between secret bit rate and distance due to the same reasons listed under Section VII-B1. However, in general, we observe that the secret bit rate in the hallway settings is low due to high bit mismatch rate.

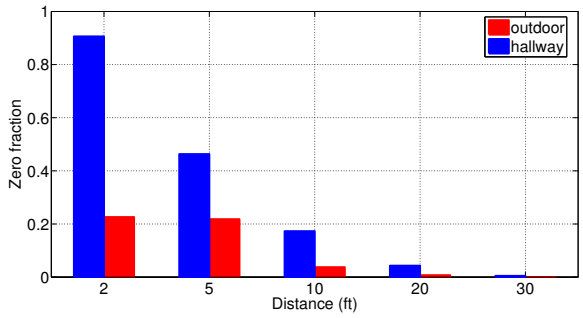

Fig. 10. Fraction of zeroes versus distance

3) Fraction of Zero-RSS values: We compare the fraction of zero-RSS values for hallway and outdoor settings in Figure 10. We find that hallway experiments have considerably large fraction of zero-RSS values in comparison to outdoor settings for all distances. This occurs because in the hallway settings, the received power mostly lies within the GRPR. A large number of zero-RSS values will lead to loss of entropy, thus reducing the secret bit rate in hallway settings. Importantly, Figure 10 also hints that we should observe higher secret bit rate under outdoor settings as we shall see in the following subsections.

4) Bit mismatch rate under outdoors with interpolation: We notice that for outdoor settings, in general, the bit mismatch rate increases with distance (Figure 11) due to the decrease in SNR with distance. We observe that under outdoor settings,

TABLE I. STANDARD DEVIATION (σ) OF RSS MEASUREMENTS AVERAGED OVER ALL THE QUANTIZATION BLOCKS.

Distance (ft)	σ dB (hallway)	σ dB (outdoor)
2	0.2739	4.6984
5	1.9664	5.3959
10	2.4059	5.2477
20	2.8009	4.4812
30	2.4344	2.9787

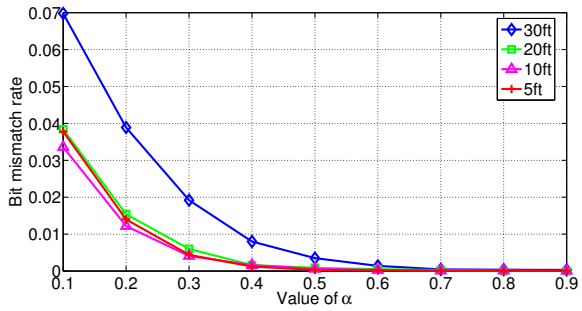

Fig. 11. Bit mismatch rate as a function of quantization parameter α for varying distances under outdoor settings (with interpolation).

the standard deviation of RSS measurements is considerably higher than the hallway settings (Table I). When the RSS variations have a small range (as in the hallways), even minor differences in the measurements of Alice and Bob could result in mismatches in their quantized bits. However, with a large variation of RSS values, we achieve significantly lower bit mismatch rates under outdoor settings as we have shown in Figure 11 (compare with Figure 8).

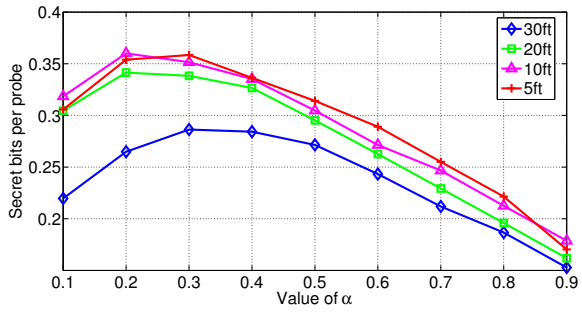

Fig. 12. Secret bit rate as a function of quantization parameter α for varying distances under outdoor settings (with interpolation).

5) Secret bit rate under outdoors (with interpolation): When we compare the secret bit rates in hallways (Figure 9) with those under outdoor settings (Figure 12), we can observe that outdoor conditions are significantly better for key extraction using Bluetooth. We can see from Figure 9 and Figure 12 that the maximum secret bit rate is only around 0.20 under hallways, whereas the secret bit rate is around 0.36 for outdoors. Since the bit mismatch rate under outdoors is lower in comparison to hallways as we have shown in Section VII-B4, we observe lower information leakage in the information reconciliation stage and consequently much higher secret bit rates in outdoor settings.

TABLE II. SECRET KEY EXTRACTION AT SMALL DISTANCES.

Metric	Indoor	Outdoor
Secret bit rate	0.0080	0.2406
Avg. no. of consecutive zero RSS values	37.6	5.91
Std. deviation of RSS values of each block	0.514	5.36

TABLE III. NIST APPROXIMATE ENTROPY TEST RESULTS

Distance (ft)	Entropy (outdoor)	Entropy (hallway)
30	0.9804	0.9861
20	0.9822	0.9863
10	0.9845	0.9840
5	0.9825	0.9782

6) Key extraction for small distances: We conduct experiments maintaining a distance of 2ft between Alice and Bob. We observe that under hallway settings the secret bit rate is very close to zero, whereas under outdoor settings the secret bit rate is around 0.24. Table II compares the results of hallway and outdoor settings. The inferior performance in hallway settings can be explained by high bit mismatch and due to very large burst of consecutive zero RSS values.

7) Entropy of secret bits: We conduct numerous NIST statistical tests to verify the randomness of the output secret bit streams generated using Bluetooth. Table III shows the per-bit entropy values for the output secret bit streams that we have extracted in our experiments; notice that all the values are almost close to one indicating that there is almost one bit of uncertainty associated with each output bit. In Table IV and Table V, we have shown the p-value for eight NIST tests, for different distances. A p-value greater than 0.01 indicates that the input bit sequence is random with a confidence of 99%. Notice that all the p-values are greater than 0.01, which verifies that all the secret bits streams are random with a very high degree of confidence.

TABLE IV. P-VALUES FROM NIST STATISTICAL TESTS FOR OUTDOOR EXPERIMENTS.

Test	30 ft	20 ft	10 ft	5 ft
Frequency	0.86	0.08	0.51	0.87
Block Frequency	0.17	0.18	0.78	0.12
Cum. sums (fwd)	0.77	0.14	0.78	0.86
Cum. sums (rev)	0.60	0.08	0.63	0.71
Runs	0.80	0.57	0.73	0.30
Longest run of 1	0.69	0.42	0.25	0.48
FFT	1.00	0.14	0.55	0.16
Approx. Entropy	0.17	0.06	0.46	0.22
Serial	0.68, 0.64	0.54, 0.60	0.50, 0.14	0.55, 0.40

TABLE V. P-VALUES FROM NIST STATISTICAL TEST FOR HALLWAY EXPERIMENTS.

Test	30 ft	20 ft	10 ft	5 ft
Frequency	0.28	0.55	0.60	0.79
Block Frequency	0.20	0.28	0.66	0.33
Cum. sums (fwd)	0.54	0.70	0.90	0.79
Cum. sums (rev)	0.39	0.29	0.61	0.55
Runs	0.23	0.40	0.55	0.97
Longest run of 1	0.48	0.21	0.92	0.89
FFT	0.96	0.86	0.73	0.97
Approx. Entropy	0.48	0.21	0.18	0.16
Serial	0.32, 0.29	0.28, 0.20	0.09, 0.01	0.76, 0.82

TABLE VI. COMPARISON OF SECRET BIT RATES - BLUETOOTH VS WIFI

Setting	Secret bits per probe
WiFi 20 ft outdoor	0.2482
Bluetooth 20 ft outdoor	0.2761
Bluetooth 30 ft outdoor	0.2079

8) Comparison of WiFi and Bluetooth secret bit rates: Premnath et al. [6] evaluated secret key extraction on Android smartphones using WiFi. We observe that the secret bit rate with WiFi under outdoor settings is comparable to the secret bit rate that we achieve with Bluetooth under similar settings (Table VI) even though Bluetooth uses lower transmit power. In their outdoor experiments, Premnath et al. used a higher transmit power of 8 dBm whereas we use a lower transmit power of 3 dBm for Bluetooth. Therefore, *Bluetooth enables power-efficient secret key extraction.*

VIII. RELATED WORK

There is a vast amount of literature on secret key extraction from time and space varying wireless wireless channels. In this section, we only cite the most relevant existing work for brevity. Premnath et al. [6] have used Android smart phones to extract secret keys from WiFi RSS measurements. Earlier, Croft et al. demonstrated their adaptive ranking-based uncorrelated bit extraction (ARUBE [17]) method using WiFi RSS measurements from Android smart phones. However, as we have demonstrated in our work, when there is heavy traffic, the WiFi sampling period could be excessively large, adversely affecting the secret key extraction performance in comparison to using Bluetooth.

Other existing works [10], [9] use low power devices like TelosB nodes for RSSI-based secret key extraction. Premnath et al. [10] have explored the use of multiple frequencies for efficient, high rate secret key extraction. Their results show that the use of multiple frequencies enables extraction of stronger secret keys. While their method uses multiple frequencies for concurrent transmissions acros *multiple* telosB nodes, Bluetooth devices that we use in our research use multiple frequencies for frequency hopping.

Recently, Mathur et al. [18] have designed a scheme to extract secret keys by observing signals from public sources such as radio or television, which does not require Alice and Bob to exchange packets during the sampling stage. However, to record symmetric measurements, in their scheme the two nodes must be less than half the wave length away, which is 6.25 cm for the 2.4 GHz range. In contrast, we evaluate our RSKE approach using Bluetooth for larger distances, up to 30 ft.

IX. CONCLUSION

We explored the use of wireless channel characteristics for establishing strong secret keys of arbitrary length between Bluetooth devices. We built and evaluated a new method, which is robust to heavy WiFi traffic, using a *very wide bandwidth* ($\gg$ 20 MHz) in conjunction with *random frequency hopping*. We implemented our method on Google Nexus One smartphones and conducted numerous experiments in indoor-hallway and outdoor settings and using real-world measurements, we showed that *outdoor settings are best* suited for secret key extraction. Furthermore, we showed that the performance of secret key generation using Bluetooth is comparable to that of WiFi while using much *lower transmit power*.

REFERENCES

[1] "Bluetooth medical & health devices," http://www.bluetooth.com/pages/medical.aspx.

[2] L. Buttyan and J.-P. Hubaux, *Security & Cooperation in Wireless Networks: Thwarting Malicious & Selfish Behavior in Age of Ubiquitous Computing*. Cambridge Univ. Press'07.

[3] M. Jakobsson and S. Wetzel, "Security weaknesses in bluetooth," in *Cryptographers' Track at RSA Conference*, 2001.

[4] D. Singelee and B. Preneel, "Security overview of bluetooth," in *Technical report COSIC*, June 2004.

[5] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments." in *MOBICOM*, 2009.

[6] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. Krishnamurthy, "Secret key extraction from wireless signal strength in real environments," *IEEE Transactions on Mobile Computing*, 2013.

[7] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *EUROCRYPT*, 1994.

[8] R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions," in *STOC '89*.

[9] 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*, 2010.

[10] S. N. Premnath, J. Croft, N. Patwari, and S. K. Kasera, "Efficient high rate secret key extraction in wireless sensor networks using collaboration," in *ACM Transactions on Sensor Networks (to appear)*, 2014.

[11] "Bluetooth 4.0," https://www.bluetooth.org/Technical/Specifications/adopted.htm.

[12] "Ieee std 802.15.1-2005," http://standards.ieee.org/findstds/standard/802.15.1-2005.html.

[13] N. Golmie, O. Rebala, and N. Chevrollier, "Bluetooth adaptive frequency hopping and scheduling," in *MILCOM'03*.

[14] "Broadcom bcm4329 product information," http://www.broadcom.com/products/Wireless-LAN/802.11-Wireless-LAN-Solutions/BCM4329.

[15] "Bluez bluetooth stack," http://www.bluez.org/.

[16] R. Morrow, *Bluetooth: Operation and Use*. New York, NY, USA: McGraw-Hill, Inc., 2002.

[17] J. Croft, N. Patwari, and S. K. Kasera, "Robust uncorrelated bit extraction methodologies for wireless sensors," in *in ACM IPSN*, 2010.

[18] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, "Proximate: proximity-based secure pairing using ambient wireless signals," in *MobiSys '11*.