Decorrelating Secret Bit Extraction via Channel Hopping in Body Area Networks

Linjia Yao*, Syed Taha Ali*, Vijay Sivaraman* and Diethelm Ostry†
*School of Electrical Engineering and Telecommunications
University of New South Wales, Sydney, Australia
Email: linjia.yao@student.unsw.edu.au; {taha, vijay}@unsw.edu.au
†ICT Centre, CSIRO, Sydney, Australia
Email: diet.ostry@csiro.au

***Abstract*—Recent research has demonstrated that two communicating parties can generate shared secret keys by exploiting characteristics of the wireless fading channel between them. These channel characteristics are symmetric, dependent on position and orientation, highly sensitive to motion, and cannot be deduced in detail by an eavesdropper. One problem with this approach, however, is that over small channel sampling intervals, successively sampled values are correlated in time, which therefore yields keys with reduced entropy. In this paper, we undertake experiments to determine the efficacy of using channel hopping to increase diversity and improve secret key entropy, in the context of body area networks.**

We conduct extensive experiments using off-the-shelf IEEE 802.15.4 devices, mounted on the human body, in a real indoor environment. Our experimental results show that: (i) channel hopping increases frequency diversity and effectively decorrelates successive channel samples, significantly increasing entropy (at minimum approximately 20%) and thereby improving the strength of the secret key, (ii) the benefit can be maximized by devising a hopping strategy that takes into account the number of channels available, the spacing between them, and the activity of the user.

I. Introduction

In recent years there has been increasing application of body area networks for healthcare and sports fitness. Wearable sensor devices such as the Toumaz Sensium [1] and FitBit [2] measure the user's vital signs (such as heart rate, temperature, blood glucose level, etc.) or profile his activity and wirelessly communicate the information to a remote basestation. Securing this date is essential given the strict ethical and legal obligations attached to medical and personal data. However, these miniature devices are severely limited in resources and public-key mechanisms prove too expensive in terms of computation, memory, and overhead, for frequent use [3]. Designing lightweight security mechanisms for these devices is still an open research area.

The problem we examine in this paper is that of secret key agreement. To secure communication, two parties need to be in possession of a shared secret. The Diffie-Hellman key exchange is typically used for this purpose, but results [4] have shown that it is not very practical for small devices. A promising alternative approach [5] is exploiting the unique properties of the wireless channel between two communicating devices. Wireless communication is fundamentally insecure but the channel between two communicating parties is unique to them and symmetric [6]. If one party, say Alice, transmits to Bob, the signal traverses multiple paths, experiencing different degrees of attenuation and phase shift, and Bob receives the summation of these multipath signals. If Bob responds in the exact same conditions, Alice's impulse response would be highly correlated to Bob's. This shared information can be used to generate a secret key. A third party, Eve, located at least half a wavelength away from either Alice or Bob, eavesdropping on all transmissions, measures a different channel, and for a dynamic multipath environment, and it is near-impossible to guess Alice-and-Bob's channel impulse response.

Physical layer secret key generation schemes typically comprise four stages: two communicating parties continuously **sample** the channel by exchanging probe packets and measuring reciprocal channel state. These channel estimations are then **quantized** to yield key bits. However, mostly due to small-scale noise effects (that are random and uncorrelated) some generated bits may disagree between the two parties, and an **information reconciliation** process corrects the mismatch by exchanging feedback. Additionally, some bits may be correlated and **privacy amplification** is used to minimize this advantage for an eavesdropper typically by employing a transform operation.

In existing work, there are limitations on how frequently Alice and Bob sample a single channel, thus limiting the secret bit generation rate. The interval over which the channel impulse response is constant is known as the *coherence time* of the channel, and if successive sample values are collected within this interval, they will be highly correlated. This has the effect of reduced entropy and weaker keys. To counter this, researchers have either prescribed compute-intensive privacy amplification protocols [7], used multiple antennas to increase path diversity [8], or sampled the channel at very slow rates [9].

We consider the fact that existing off-the-shelf radio devices can be enabled to communicate over a number of discrete channels (14 for WiFi, 16 for 802.15.4), and this ability can be leveraged to improve diversity. Sampling multiple channels over a small time interval can be considered analogous to sampling different communication paths. This should decorrelate successive key bits, thereby improving entropy without sacrificing bit rate. In this case the wider apart the channels are spaced, the more diverse and uncorrelated the paths will be. On the other hand, the more discrete channels that are available, the greater the number of paths, allowing for greater temporal diversity, i.e. successive samples taken on a single channel will be spaced wider apart in time, ideally a period greater than the coherence time of the channel. In practice, however, bandwidth is limited and there is a tradeoff between channel spacing and the number of channels available. If the channel spacing is too small, the paths will not be very diverse, and if samples on a single channel are timed too close together, existing paths will be reused. In both cases, sample values will be correlated.

The coherence time, typically in the order of milliseconds, we note, is also highly responsive to user activity, and this affects key entropy. Our results indicate that if the user is very active, even sampling on a very small number of channels will give good entropy because the communication paths change very rapidly. For low activity, however, where paths are slow to change it becomes important to sample intelligently on different channels while trying to maximize channel spacing.

Our contributions in this paper are:

- We perform extensive experiments using bodyworn devices to measure the efficacy of channel hopping in decorrelating secret bits and improving key entropy. Our results, using the approximate entropy metric, show an increase of up to 20% in some cases.
- Our results further indicate that this advantage can be maximized by devising an intelligent hopping strategy that takes into account the number of channels available to the communicating parties, the spacing between them, and the activity of the user.

This paper is organized as follows: in Section II, we discuss prior work. Section III describes our experimental setup and highlights the effect of channel hopping on decorrelation of samples. In Section IV, we present key generation results for different hopping strategies and compare for varied user activity. We conclude in Section V.

II. Prior work

Recent years have witnessed growing interest in exploiting wireless channel characteristics to generate shared secrets. A theoretical foundation for this approach is laid in [6] and [10] which suggest that two parties can generate shared secrets using correlated random variables in the presence of an eavesdropper. Wireless channel characteristics used for key generation include signal phase, time delay, angle of arrival and received signal strength(RSS). RSS is commonly used because it is easy to obtain on most off-the-shelf radio devices.

In [11], the authors describe a method to extract key bits from a statistical Gaussian channel. They validate this mechanism using WiFi in an indoor environment and generate bits at a rate of 1 bit/sec with very high bit agreement at both ends. This approach is extended in [12], where the authors gather empirical measurements of RSS over a single channel for various environments using laptops with WiFi hardware and propose a multi-bit quantization method. The authors use privacy amplification to strengthen the generated keys.

In [13], the authors aim for a very high bit generation rate of 40 bits/s over a single channel using TelosB motes. A transform operation is used especially to decorrelate secret key bits, adding to the computational complexity of the process.

In [14], the authors demonstrate an RSS-based key generation scheme for body area networks where the Savitzky Golay filter is used to isolate different frequency components of the dynamic wireless channel, highlighting their individual contribution to key generation. Temporal diversity is ensured by sampling the wireless channel at very low rates. On the other hand, [8] increase secret key entropy by using multiple-antenna devices, thereby increasing the number of statistically independent communication paths available.

All of the aforementioned works expressly rely on device movement to create path diversity, giving rise to RSS channel fluctuation which can be quantized to produce secret bits. In [15], authors propose exploiting dynamic physical environments, where human movements create the necessary fluctuations.

To the best of our knowledge, only one work in the literature employs channel hopping: in [5] the authors investigate how hopping may provide path diversity and channel variation in a purely static scenario. Their results demonstrate that basic channel hopping is a good source for correlated randomness for the two endpoints, but the amount of meaningful information that can be extracted is limited because the channel is very slow to decay in a static setting, and cannot be reused often.

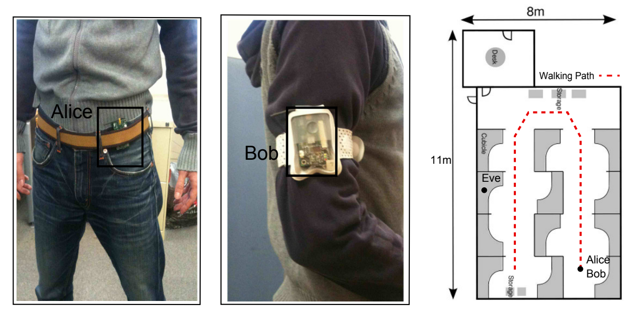
Our work differs in that we study hopping in a dynamic scenario where, our results indicate, the *sequence* in which hopping is performed becomes very important. Channel hopping effectively decorrelates samples very closely spaced in time, but, as we argue in this paper, the effect is strongly influenced by user activity, and can be maximized by intelligently choosing between the number of channels available and the inter-channel spacing.

III. Channel Spacing and Correlation of Successive Samples

In this section we examine how correlation between successive samples varies with channel spacing.

A. Experimental Setup

We conducted experiments using MicaZ motes, with 16 channels (channel 11 to channel 26) operating between 2.405 to 2.480 GHz frequency range with steps of 5 MHz. Experiments were performed in an indoor office space consisting of multiple cubicles and a pair of sensors motes(Alice and Bob) were mounted on the subject's right arm and left waist as shown in Fig. 1. Both sensors were configured to transmit probe packets at 0 dBm power at a rate of 50 packet/second. During the experiment, the subject walked around the office



(a) Alice on user's waist (b) Bob on user's right arm (c) Office Layout

Fig. 1: Bodyworn sensors mounted on male subject and layout of office environment with Eve's location and subject's walking path.

at about 1 m/s, the experiment lasted 40 minutes, and yielded approximately 120,000 RSS values per sensor mote.

Channel hopping pattern is as follows: packets are sent in both directions of the link every 20 ms, enabling endpoints to sample the RSS reading. Each exchange is followed by a reference exchange which uses channel 26 as a baseline for comparison. Alice and Bob's sampling order is therefore as follows: probe on channel *i*, probe on channel 26, probe on channel *i+1*, probe on channel 26, and so on. This ensures that sample values all have the same delay and the same baseline reference (channel 26) and channel spacing incrementally hops by one channel every transmission, as illustrated in Fig. 2.

B. Correlation vs. Channel Spacing

To evaluate the correlation of successive sampling values, we employ the Pearson correlation coefficient r:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \qquad (1)$$

where X_i and Y_i are the RSSI values of the reference packets and successive packets on a different channel. The correlation coefficient r returns a value in [-1,1], where -1 and 1 indicates anti-correlation and perfect correlation respectively, and 0 indicates no correlation.

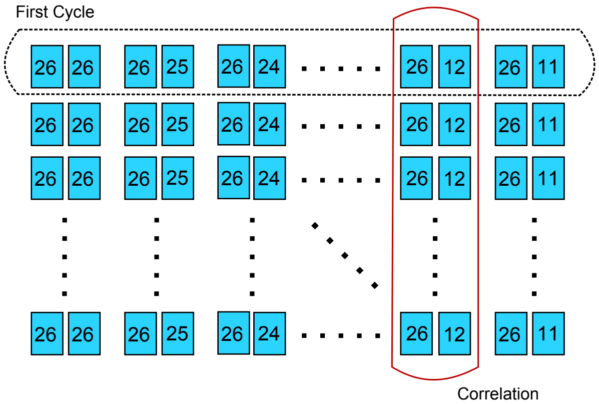


Fig. 3: Probe packets received by an endpoint for correlation analysis.

During the experiment, each sensor records a matrix of RSS values as shown in Fig. 3. Here, each row represents one sampling cycle with comparing channel hopping from channel 26 to 11, while in a column, sample values can be divided into 16 groups, each comprising a value from a reference exchange (channel 26) and the successive value where the channel is incrementally varied.

Taking two groups of column values in Fig. 4, for example, adjacent sampling values are seen to be highly correlated without hopping (i.e. both samples are taken for channel 26), as compared to the case for successive sample values collected when hopping from channel 26 and channel 11 in Fig. 4b, which show greater variation.

This trend is also seen when correlation of successive samples is plotted against channel spacing in Fig. 5, i.e. the correlation decreases as channel spacing between successive samples increases. Without channel hopping, successive values have a correlation coefficient of about 0.75. For the same sampling delay, if successive values are collected over a channel spacing of at least 7 channels (i.e. 35MHz), the correlation coefficient falls to less than 0.5, a significant drop. However, we observe that the correlation cannot be reduced to zero even the channel spacing is maximized (a spacing of 15, i.e. 75 MHz). This is due to the slow fading component in the RSS profile, which has been noted by other researchers too [11], [14], when sampling over a single channel. [11] specifically has suggested using a moving average filter to

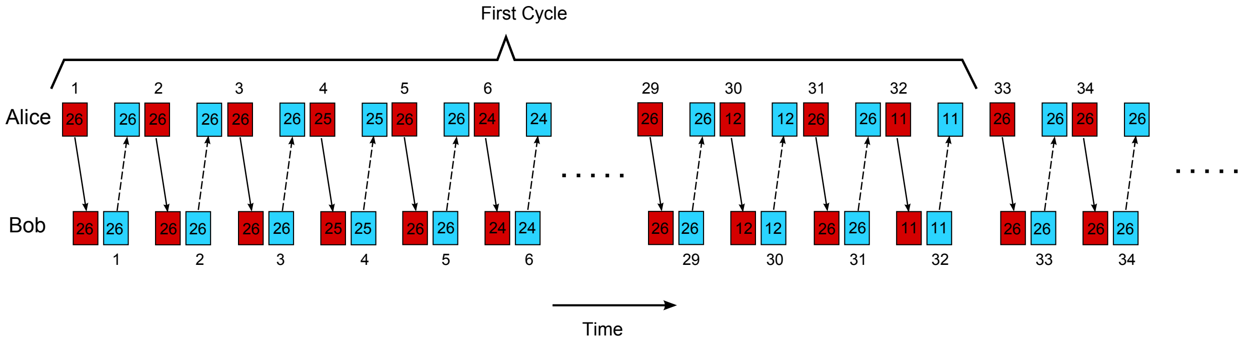
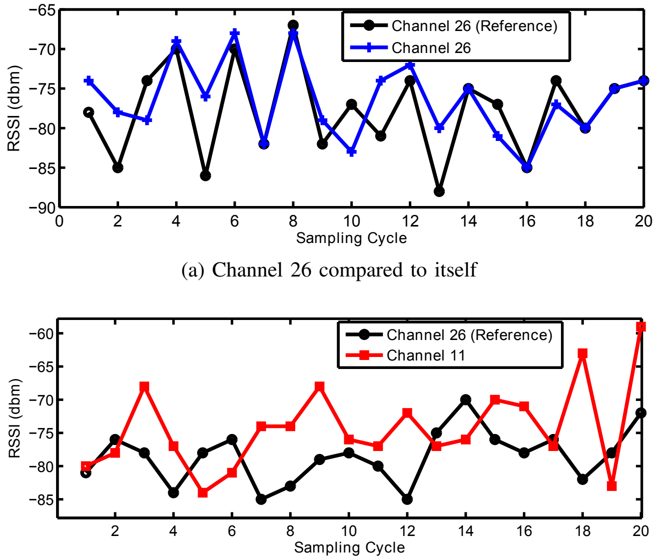


Fig. 2: Channel hopping sequence: Alice and Bob exchange probe packets at 50 pkts/sec, all channels are sampled incrementally, and channel 26 is the reference for each exchange.



(a) Channel 26 compared to itself

(b) Channel 11 compared to channel 26 (reference channel)

Fig. 4: RSSI values of adjacent sampling

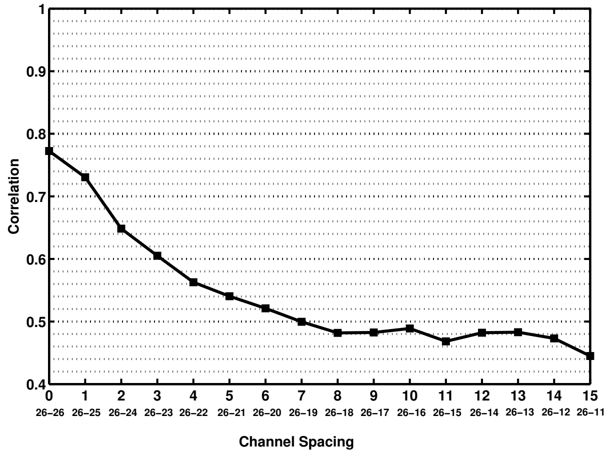


Fig. 5: Correlation of adjacent sampling with varying channel spacing

remove this component. However, in body area networks, the design and parameters of this filter are highly dependent on device placement and user activity, and we choose to address this question in future work.

IV. Secret Key Generation

In this section, we evaluate the randomness of the secret bits generated using different hopping strategies for varied user activity.

A. *Experimental Setup*

Our MicaZ motes were configured as described earlier in Section III. We conducted two sets of experiments: in *High Activity*, the user walked continuously around the office space at about 1 m/s, while for *Low Activity*, the user sat at his desk and worked on his computer. A third party Eve was introduced to eavesdrop on communication between Alice and Bob, positioned as shown in Fig. 1, stationary on a desk and always more than half wavelength(roughly $6.25cm$) away from the legitimate parties, Alice and Bob, which were mounted on the subject's waist and arm respectively. Overall, we collected approximately 280,000 RSS values for each mote in *High Activity* and approximately 180,000 RSS values in *Low Activity*.

TABLE I: Channel Hopping Strategies

Spacing	Channel Hopping Sequence
0	26-26-26-26 ...
1	26-25-24-23 ... 13-12-11-12 ... 24-25-26-25-24 ...
2	26-24-22-20-18-16-14-12-14-16-18-20-22-24-26 ...
4	26-22-18-14-18-22-26-22 ...
7	26-17-24-15-22-13-20-11-18-25-16-23-14-21-12-19-26-17 ...
8	26-18-26-18 ...
15	26-11-26-11 ...

B. *Channel Hopping Strategies*

We tried seven hopping strategies with spacing as shown in Table I, for both modes of activity. As noted earlier, the maximum channel spacing is 15 channels. A spacing of 0 equates to the default single channel sampling case. A spacing of 1 means that the devices will incrementally cycle through all the channels in turn from channel 26 to channel 11 and then cycle back from there. For a spacing of 4, the sequence will be channel 26, channel 22, channel 18, channel 14, and then cycling back, channel 18, channel 22, channel 26, and so on. A spacing of 15 indicates that hopping back and forth between two channels, channel 26 and channel 11.

A spacing of 7 is interesting for two reasons: as observed in Fig. 5, the correlation between successive values falls to near minimum from 0.7 to 0.5 when the channel spacing is 7 and greater. Also, 7 is not a factor of 16, and if the channel count is allowed to overflow, it will take a full 16 sampling intervals for the device to cycle back to a single channel. The sequence is listed in Table I. The hopping sequence is computed by the formula:

$$channel_{i+1} = ((channel_i - 11 + 7) mod 16) + 11 \qquad (2)$$

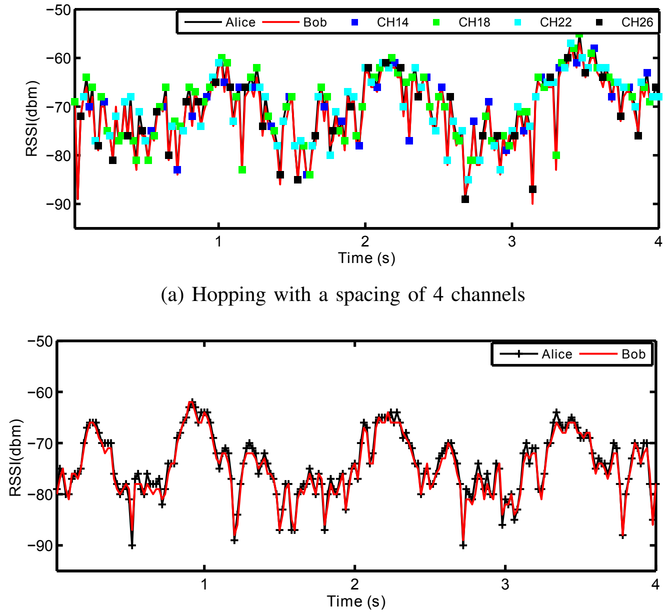
where $channel_i$ is the current channel the device is on. This strategy ensures all 16 channels are individually sampled in a cycle and successive channels are separated by about 35MHz.

C. *Quantization*

To convert the RSS values into binary key bits, we use the quantizer developed in [8]. This quantizer has multiple quantization levels and guard bands, and produces equiprobable output. Quantization intervals and guard band values are determined by:

$$\int_{q_{i-1}}^{q_i - g_i} f_{\widetilde{h}} d\widetilde{h} = \frac{1-\alpha}{m} \quad ; \quad \int_{q_i - g_i}^{q_i} f_{\widetilde{h}} d\widetilde{h} = \frac{m}{1-\alpha} \qquad (3)$$

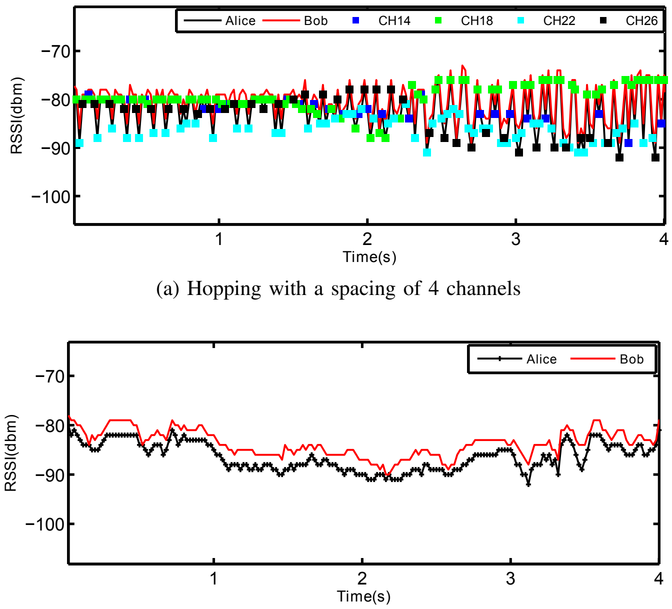
where q_i and g_i are the quantization and guard band thresholds respectively, α is the ratio of sampling values that



(a) Hopping with a spacing of 4 channels

(b) Sampling over a single channel (no hopping)

Fig. 6: RSSI of the probe packets exchanged in *High Activity*



(a) Hopping with a spacing of 4 channels

(b) Sampling over a single channel (no hopping)

Fig. 7: RSSI of the probe packets exchanged in *Low Activity*

are discarded, m is the number of discrete quantization values, and $\widetilde{h}$ is the sampling values with probability distribution $f_{\widetilde{h}}$.

For each hopping pattern, RSS values are separated as per their channel, quantized separately, i.e. there are individual bit-strings produced for each channel, which are then interleaved into a composite bit stream as per their sampling order.

D. Results

Figs. 6 and 7 show traces of RSS profile for *High* and *Low Activity* for two hopping strategies, the first having a channel spacing of 4 channels and compared with sampling over a single channel without hopping. It is clear that greater variation in RSS can be achieved by alternately sampling over different channels than over one alone.

We use the approximate entropy metric [16] to evaluate the randomness of the secret keys thus produced. The approximate entropy test returns a value between 0 and 1 where higher values indicate a more random bit stream.

As shown in Fig. 8, for *High Activity*, the approximate entropy of secret key bit streams is below 0.75 for a single channel case, and it increases to over 0.9 using a hopping strategy with spacing 8 and over, an approximate 20% improvement. In this case, due to constant user activity, the channel coherence time is very small and the best strategy, therefore, is to sample a minimal number of channels with a suitably large spacing.

For the *Low Activity* scenario, shown in Fig. 8, single channel sampling again has the lowest approximate entropy value at approximately 0.3. This is because the relatively slow rate of movement causes communication paths to decay far more slowly and successive sample values are highly correlated. Approximate entropy increases with greater channel spacing but only till a spacing of 7 channels is reached after which it falls. This is interesting because it highlights the essential tradeoff between spacing and number of channels: as spacing increases from 0 to 1, there is a dramatic increase in entropy (from approximately 0.3 to 0.8) because even though the channels are closely spaced, temporal diversity is much greater, i.e. it takes a full 16 sampling intervals before the same channel is sampled again. As spacing increases however, temporal diversity is reduced: at a spacing of 8, the time between sample values on the same channel is only two sample intervals, and the samples are therefore more correlated. This trend is more evident if we consider Fig. 9 where we plot approximate entropy against the number of channels used for hopping. For *Low Activity*, entropy generally increases as a larger number of channels is sampled. A channel spacing of 7 in this case, will sample each of the available channels with sufficient temporal and frequency diversity to maximize the entropy. This example highlights the importance of choosing the right hopping strategy to maximize entropy, taking into account user activity, the number of channels available, and the inter-channel spacing.

Tables II and III show bit generation rate and mismatch for Alice and Bob and Eve for *High* and *Low Activity* scenarios.

We assume that Eve knows the channel hopping sequence, the quantization algorithm and parameter settings, and can capture the probe packets exchanged by Alice and Bob. As can be seen from Table II and Table III, most of the cases resulted in approximately a 50% secret key bit mismatch for Eve which means that her chances of guessing the bits generated by Alice and Bob are equivalent to guessing for a fair coin toss, which is the ideal scenario. In terms of bit generation rate and bit mismatch for Eve, channel hopping on the part of Alice and Bob consistently shows comparable or superior performance.

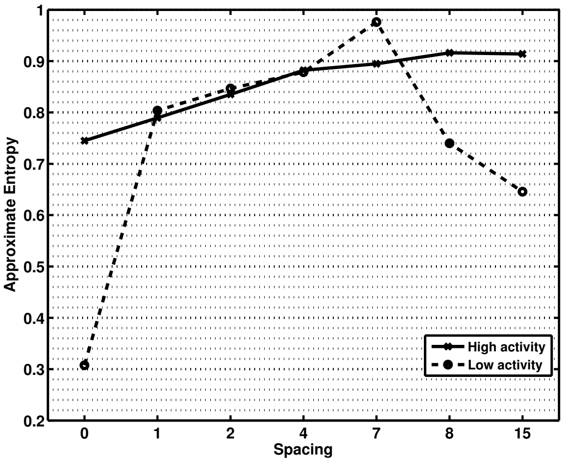


Fig. 8: Approximate entropy of bit streams vs. channel spacing

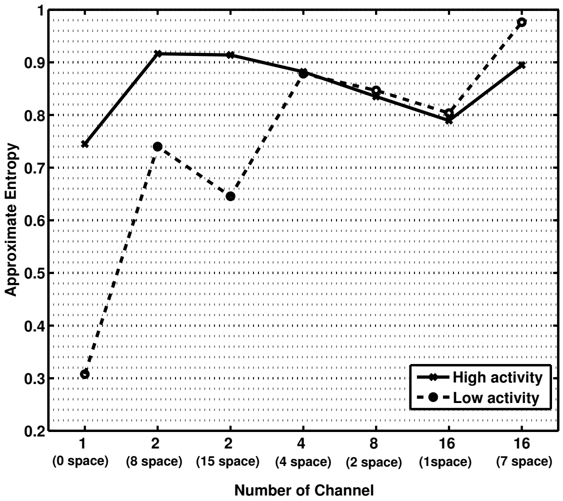


Fig. 9: Approximate entropy of bit streams vs. number of channels sampled

TABLE II: *High Activity*: Key generation performance for different hopping strategies (quantizer settings: $m = 2$, $\alpha = 0.3$.)

Space	Key Rate (bits/s)	Alice and Bob: Bit Mismatch	Eve: Bit Mismatch
0	75.96	2.68%	50.95%
1	70.34	2.02%	49.96%
2	71.41	2.12%	52.67%
4	68.44	2.03%	50.52%
7	70.69	1.95%	50.16%
8	76.54	2.01%	51.76%
15	71.92	2.53%	51.41%

TABLE III: *Low Activity*: Key generation performance for different hopping strategies (quantizer settings: $m = 2$, $\alpha = 0.3$.)

Space	Key Rate (bits/s)	Alice and Bob: Key Mismatch	Eve: Key Mismatch
0	92.62	6.30%	34.89%
1	65.80	6.59%	49.00%
2	53.13	4.95%	51.25%
4	52.74	5.15%	45.55%
7	67.72	5.55%	48.06%
8	60.59	2.04%	50.00%
15	63.07	4.08%	47.29%

V. Conclusion

In this paper, we study the efficacy of the channel hopping technique in decorrelating successive sample values for secret key generation using the wireless channel. Furthermore, we identify key parameters affecting performance, namely the channel spacing, the number of available channels, and user activity. Our results show that it is possible to devise a hopping strategy to maximize the benefit by trading off between channel spacing and number of channels to maintain high temporal and frequency diversity. Our experiments indicate a minimum of 20% increase in key entropy and improved performance if channel hopping is used. A more systematic study on intelligent channel hopping is left for future work.

References

[1] Sensium life platform. Toumaz Technology Ltd. [Online]. Available: http://www.toumaz.com

[2] Fitbit ultra technology. Fitbit Inc. [Online]. Available: http://www.fitbit.com

[3] H. E. V. G. A.S. Wander, N. Gura and S. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” in *IEEE PerCom*, 2005.

[4] E. Blass and M. Zitterbart, “Efficient Implementation of Elliptic Curve Cryptography for Wireless Sensor Networks,” Universität Karlsruhe, Tech. Rep., 2005.

[5] M. Wilhelm, I. Martinovic, and J. B. Schmitt, “Secret keys from entangled sensor motes: implementation and analysis,” in *ACM WiSec’10.*

[6] U. M. Maurer, “Secret key agreement by public discussion from common information,” *IEEE Transactions on Information Theory*, 1993.

[7] S. J. N. Patwari, J. Croft and S. Kasera, “High-rate uncorrelated bit extraction for shared secret key generation from channel measurements,” *IEEE Transactions on Mobile Computing, 9(1)*, 2010.

[8] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, “Exploiting multiple-antenna diversity for shared secret key generation in wireless networks,” in *IEEE INFOCOM*, 2010.

[9] S. T. Ali, V. Sivaraman, and D. Ostry, “Zero reconciliation secret key generation for body-worn health monitoring devices,” in *ACM WiSec’12.*

[10] U. Maurer and S. Wolf, “Secret-key agreement over unauthenticated public channel,” *Information Theory, IEEE Transactions on*, 2003.

[11] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, “Radio-telepathy: extracting a secret key from an unauthenticated wireless channel,” in *ACM MobiCom*, 2008.

[12] S. Jana, S. N. Premnath, M. Clark, S. Kasera, N. Patwari, and S. Krishnamurthy, “On the effectiveness of secret key extraction from wireless signal strength in real environments,” in *ACM MobiCom*, 2009.

[13] J. Croft, N. Patwari, and S. Kasera, “Robust uncorrelated bit extraction methodologies for wireless sensors,” in *ACM/IEEE*, ser. IPSN, 2010.

[14] S. T. Ali, V. Sivaraman, and D. Ostry, “Secret key generation rate vs. reconciliation cost using wireless channel characteristics in body area networks,” in *IEEE Trustcom*, 2010.

[15] I. M. P. Barsocchi, S. Chessa and G. Oligeri, “A cyber-physical approach to secret key generation in smart environments,” *Journal of Ambient Intelligence and Humanized Computing*, 2011.

[16] NIST, “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” 2010.