Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel

Conference Paper · January 2008

DOI: 10.1145/1409944.1409960 · Source: DBLP

5 authors, including:

W. Trappe
Rutgers, The State University of New Jersey
281 PUBLICATIONS **10,784** CITATIONS
SEE PROFILE

Narayan Mandayam
Rutgers, The State University of New Jersey
280 PUBLICATIONS **10,663** CITATIONS
SEE PROFILE

Radio-telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel

Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye[†], Alex Reznik[†]
WINLAB, Rutgers University, North Brunswick, NJ, USA
[†]InterDigital, King of Prussia, PA, USA
{suhas, trappe, narayan}@winlab.rutgers.edu,
[†]{chunxuan.ye, alex.reznik}@interdigital.com

ABSTRACT

Securing communications requires the establishment of cryptographic keys, which is challenging in mobile scenarios where a key management infrastructure is not always present. In this paper, we present a protocol that allows two users to establish a common cryptographic key by exploiting special properties of the wireless channel: the underlying channel response between any two parties is unique and decorrelates rapidly in space. The established key can then be used to support security services (such as encryption) between two users. Our algorithm uses level-crossings and quantization to extract bits from correlated stochastic processes. The resulting protocol resists cryptanalysis by an eavesdropping adversary and a spoofing attack by an active adversary without requiring an authenticated channel, as is typically assumed in prior information-theoretic key establishment schemes. We evaluate our algorithm through theoretical and numerical studies, and provide validation through two complementary experimental studies. First, we use an 802.11 development platform with customized logic that extracts raw channel impulse response data from the preamble of a format-compliant 802.11a packet. We show that it is possible to practically achieve key establishment rates of ~ 1 bit/sec in a real, indoor wireless environment. To illustrate the generality of our method, we show that our approach is equally applicable to per-packet coarse signal strength measurements using off-the-shelf 802.11 hardware.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Distributed networks, Wireless communication

General Terms

Security, Algorithms, Measurement

MobiCom'08, September 14–19, 2008, San Francisco, California, USA.
Copyright 2008 ACM 978-1-60558-096-8/08/09 ...$5.00.

1. INTRODUCTION

Traditional network security mechanisms rely upon cryptographic keys to support confidentiality and authentication services. However, in a dynamic mobile wireless environment, with peer-to-peer associations being formed on-the-fly between mobile entities, it is difficult to ensure availability of a certificate authority or a key management center. Since such scenarios are likely to become more prevalent, it is necessary to have alternatives for establishing keys between wireless peers without resorting to a fixed infrastructure.

In this paper, we explore an alternative for building cryptographic services by exploiting an untapped resource – the wireless channel itself. The specificity of the radio channel between two wireless devices, and its rapid decorrelation with distance, provide a basis for the creation of shared secret information, such as cryptographic keys, even in the presence of an eavesdropper. In typical multipath environments, the wireless channel between two users, Alice and Bob, produces a time-varying, stochastic mapping between the transmitted and received signals. This mapping is both, location-specific and reciprocal, i.e., the mapping is the same whether Alice is the transmitter with Bob as the receiver or vice-versa. The time-varying mapping, commonly termed *fading*, decorrelates over distances of the order of half a wavelength, λ. Thus, an adversary, Eve, who is more than $\lambda/2$ away from both Alice and Bob, experiences fading channels to Alice and to Bob that are statistically independent of the fading between Alice and Bob. These properties allow us to generate a common, secret cryptographic key at Alice and Bob such that Eve gets no information about the generated key. For example, at 2.4 GHz, we only require that Eve be roughly $\lambda/2 = 6.25$ cm away from Alice and Bob to ensure that she gets no useful information. Thus, while fading is typically considered harmful, we profitably exploit it to extract perfectly secret bits without leaking information to an adversary.

The extraction of secret bits from the wireless channel can be viewed as a 'black-box' that can be advantageous in various ways, putting to good use information that is already available from the channel. For example, in the current 802.11i standard, session keys for communication between a station and an AP are derived by hashing together authentication credentials and nonces exchanged in the clear. This ties the confidentiality of future messages to the authentication credentials, and if these credentials are ever compromised then an adversary will be able to derive the session keys and decrypt past encrypted messages. If the nonces can be derived in an information-theoretically secret manner from the channel between two users, then a passive adversary has no means to derive the session keys even if it learns the authentication credentials [1]. Further, session keys can be updated using these secret bits derived from the channel, instead of relying on previously existing keys [1], thus ensuring that the confidentiality of each new session is protected independently of earlier sessions.

Yet another vulnerability in 802.11i stems from the fact that during the establishment of a secure link between a station and an AP, all messages exchanged over the air, including management frames, are sent unencrypted until both parties have obtained the session key (c.f. the *temporal key* (TK) in 802.11) and are therefore susceptible to eavesdropping and to spoofing by other users. While the 802.11w amendment seeks to protect some management frames from such attacks, it too fails to protect messages exchanged before the the establishment of TKs. Unfortunately, securing the initial exchanges between the parties requires them to share a key that is not established until later. Our key extraction mechanism provides a natural solution by allowing the parties to generate a temporary key that protects the interim exchanges before the formal keys are in place.

Ad hoc or peer-to-peer networks present another avenue where our technique can be useful. Alice may not care to establish Bob's identity if she merely wishes to employ his forwarding services. In such a scenario, she may nevertheless wish to establish a confidential link with Bob by using the channel to form a key prior to encrypting subsequent data, thereby preventing eavesdropping.

Prior work in information theory has noted the potential of using the wireless channel for generating shared secret bits, but most of this work has been aimed at computing theoretical limits and has not provided practical algorithms, nor a demonstrable and quantifiable impact on security. Our contribution in this paper is: (1) We translate prior information-theoretic ideas into a practical protocol applied to wireless channels, (2) we build a new algorithm for key extraction that, unlike prior schemes, does not require an authenticated channel, and study performance for typical fading, and (3) we validate our algorithm using channel impulse responses measured using the 802.11a packet preamble on a customized FPGA-based 802.11 platform and a second study that uses only coarse per-packet RSSI information readily available to off-the-shelf 802.11 platforms.

Existing mobile radio platforms already provide the information we need, but such data are normally discarded after physical layer processing and can be profitably exploited to benefit security. The approach we present augments, rather than replaces existing cryptographic security mechanisms– it provides a new approach to establishing keys that is useful when there is no key management infrastructure. In Section 2 we summarize the related work, in Section 3 we describe our system model and the design issues relevant to our problem, in Section 4 we describe our key-extraction algorithm in detail, in Section 5 we evaluate its performance and in Section 6 we present two experimental studies that validate our algorithm on 802.11a hardware. We present a discussion on the tradeoffs and security of our key-extraction method in Section 7 and we conclude in Section 8.

2. RELATED WORK

Information-theoretic literature has explored the use of information from the physical layer in deriving security benefits. In [2,3], the authors introduced the problem of generating identical bits based on correlated information available to two users such that a third eavesdropping user does not learn anything about the generated key. They showed, provided Alice and Bob already share an authenticated public channel, that it is possible to generate identical keys at the two users. The standard method for generating secret keys at Alice and Bob consists of three basic steps and has been utilized by a number of proposed systems [4–6]. In *advantage distillation* [2, 7], the legitimate users, Alice and Bob, obtain correlated information while Eve is allowed to eavesdrop, so that Alice & Bob share greater information[1] than that shared between Alice & Eve or Bob & Eve. Alice and Bob then convert their information into bits. In the *information reconciliation* stage [5], Alice and Bob exchange error-correcting messages over an authenticated public channel that allow them to agree on an identical string of bits. However, the publicly exchanged messages reveal a certain amount of information about the bit strings to Eve. In *privacy amplification* [9], Alice and Bob diminish the partial information revealed to Eve by systematically discarding some of their common bits. Efficient protocols have since been designed [5, 10][2] to allow key generation without leaking information to an eavesdropping adversary.

A central assumption in this entire body of work is that Alice and Bob have an *authenticated channel available to them* even before key generation begins. This is an unrealistic assumption in practice because the availability of an authenticated channel implies that Alice and Bob already share a secret key to begin with! Therefore, the purpose of generating a common secret key is defeated.

In [12], Maurer and Wolf showed that secret key extraction without an authenticated channel is possible only if Eve cannot possibly transmit a signal to Bob that is statistically indistinguishable from signals coming from Alice (and vice-versa). This provides an important insight that has not been translated into a practical algorithm. Our work is the first to build upon this result: we use the wireless channel to guarantee that Eve does not possess the required information to prevent key generation.

More recently, [13] examined PHY-layer based authentication and confidentiality in wireless systems. The work in [14, 15] looked at authentication using channel signatures between the transmitter and receiver(s). Our work is perhaps most closely related to [16], which proposes a scheme for generating secret bits from correlated observations of deep fades by two users communicating via a TDD link. This work focuses on the theoretical construction for extracting randomness through universal hash families. However, they do not demonstrate or evaluate the amenability of the wireless channel to detection of deep fades by both users, nor the precision needed in the TDD process for their scheme. A quantification of the secret key rate versus parameters associated with the underlying fading process or parameters involved in their algorithm was not provided. Additionally, we note that their method focuses primarily on a passive adversary. The reliance on deep fades may be exploited by an active adversary that produces greater interference power at one legitimate user than the other so

[1] The amount of information between two observations X and Y is measured by the *mutual information* $I(X; Y)$ [8].

[2] Much of this work was done in the context of quantum key distribution [11].

that a deep fade for one user may not be a deep fade for the other. In [17], a method exploiting channel reciprocity using ultra-wideband (UWB) channels to generate secret bits was presented. In [18], specialized electronically steerable antennas were proposed for use in generating key bits by exploiting channel reciprocity. The methods in [16–18] all rely on conventional reconciliation for correcting bit-errors, and thus require an authenticated channel. In [19, 20], a method for secret key generation based on phase *reciprocity* of frequency selective fading channels was proposed. While this is attractive, it is difficult to implement as accurate phase information is hard to harvest from existing platforms.

In contrast to prior work, our algorithm transcends the requirement of an authenticated channel, does not require specialized hardware and is not limited to UWB channels. We provide a fundamental analysis between the performance of our scheme and underlying parameters governing fading and quantization. Further, we provide two real implementations of our scheme and show that existing mobile platforms already provide sufficient information for producing secret bits. We evaluate the randomness of the bit-sequences produced by our algorithm, a generally overlooked aspect in prior work on secret key generation, and show that they are suitable for use as cryptographic keys. Lastly, we note that our technique may be compared with classical key establishment techniques such as Diffie-Hellman, which also use message exchanges to establish keys. However these rely upon unproven arguments of computational hardness of problems such as the discrete logarithm problem or factoring a product of large prime numbers. Our algorithm, though, provides information-theoretic secrecy, does not assume bounded computation power at the adversary and further, represents practical methods to achieve this type of security. The cost of enabling unconditional security must be borne out in some form – in our case this may take the form of collecting correlated information by probing – but in fact, depending upon how our method is used, much of the required information is already available in present day systems. In this way we provide a means to realize in wireless networks the same benefits that quantum cryptography has enabled using optical fiber links.

3. SYSTEM MODEL & DESIGN ISSUES

The crucial insight that allows the wireless channel to be amenable for generating a secret key is that the received signal at the receiver is modified by the channel in a manner that is unique to the transmitter-receiver pair. This distortion depends critically upon the location of the transmitter, the receiver, and scatterers. Typically, such distortion is estimated at the physical layer of the receiver and dealt with for reliable physical layer decoding. Since this information is always present and uniquely corresponds to the transmitter-receiver pair, it also provides our transmitter (Alice) and receiver (Bob) a means to *privately* establish secret bits. We now focus on the challenges of using the stochastic nature of the wireless channel to secretly establish bits. We break down our discussion to include a description of: (1) the underlying channel model associated with multipath fading; (2) the tools needed to obtain bits from the channel response; and (3) the design goals that need to be addressed in order to reliably establish these bits. To assist the reader, we provide notation in Table 1. We assume an attacker that can either act as an eavesdropper or who may inject messages to impersonate Alice or Bob. We present further considerations of adversarial actions in Section 7.

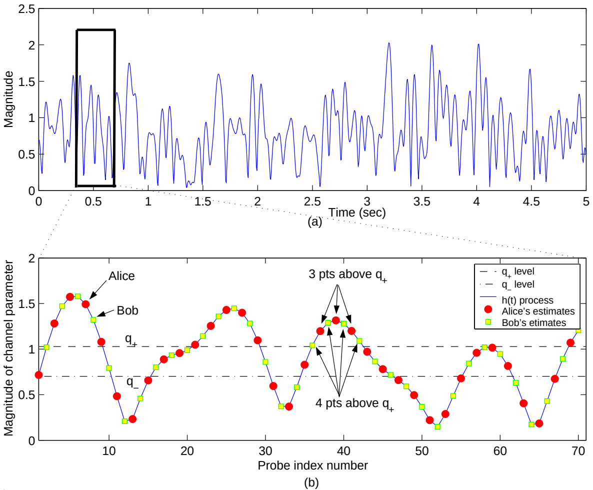


Figure 1: (a) A sample realization of a Rayleigh fading stochastic process. (b) Successive channel estimates of the process by Alice and Bob showing excursions above the q_+ and below the q_- levels on a magnified portion of (a).

Symbol	Meaning
$\mathbf{h}$	Stochastic channel parameter of interest
$h(t)$	Value of the stochastic process $\mathbf{h}$ at time t
$s(t)$	Probe signal transmitted to estimate $h(t)$
f_d	Maximum Doppler frequency (Hz)
f_s	Rate at which each user sends probes (Hz)
q_+, q_-	Quantizer bin boundaries (Upper and lower resp.)
m	Reqd. min. # of estimates in a excursion
N	Length of key in bits
R_k	Rate of generation of secret bits (s-bits/sec)
p_e	Probability of a bit error
p_k	Probability of key mismatch $= 1 - (1 - p_e)^N$

Table 1: A summary of the notation used

3.1 Channel model

Let $h(t)$ be a stochastic process corresponding to a time-varying parameter that describes the wireless channel between Alice and Bob. Although there are many choices for $h(t)$, for our discussion, we shall assume that $h(t)$ is the magnitude of the transfer function of the multipath fading channel between Alice and Bob evaluated at a fixed *test frequency*, f_0. Implicit in this formulation is the observation that the system transfer function of the channel is the same in the Alice →Bob direction as in the Bob→Alice direction *at a given instant of time*. This follows from reciprocity, which is a fundamental property of electromagnetic wave propagation [21] in a medium and must not be confused with additive noise or interference, which may be different for different receivers. To distinguish between the channel parameter of interest, and its value at a given time, we denote the parameter by $\mathbf{h}$ and refer to its value as $h(t)$. To estimate the parameter $\mathbf{h}$, Alice and Bob must transmit known probe signals to one another. Each party can then use the received signal along with the probe signal to compute an estimate $\hat{h}$ of $\mathbf{h}$. Since practical radios are *half duplex* due to hardware constraints, Alice must wait to receive a probe signal from Bob before she can transmit a probe to him and vice-versa. In the time between the two successive probes, $h(t)$ changes slightly in a manner that is modeled by an appropriate probability distribution. The received signal at Alice and Bob due to successive probes may be written as

$$r_a(t_1) = s(t_1)h(t_1) + n_a(t_1) \quad (1)$$
$$r_b(t_2) = s(t_2)h(t_2) + n_b(t_2), \quad (2)$$

where $s(t)$ is the known probe signal, n_a & n_b are the independent noise processes at Alice and Bob and t_1 & t_2 are the time instants at which successive probes are received by Alice and by Bob, respectively. Using the received signal, Alice and Bob, each compute (noisy) estimates of $\mathbf{h}$:

$$\hat{h}_a(t_1) = h(t_1) + z_a(t_1) \quad (3)$$
$$\hat{h}_b(t_2) = h(t_2) + z_b(t_2), \quad (4)$$

where z_a and z_b represent the noise terms due to n_a and n_b after processing by the function that estimates $\mathbf{h}$. We refer the reader to [22] for designing good estimators for $\mathbf{h}$. The estimates $\hat{h}_a$ and $\hat{h}_b$ are in all likelihood unequal, due in part to the independent noise terms and in part to the time lag τ. However they can be highly correlated if Alice and Bob send probes to one another at a fast enough[3] rate, i.e. if $\tau = t_2 - t_1$ is small. By repeatedly sending probes in an alternating manner over the time-varying channel, Alice and Bob can generate a sequence of n estimates $\underline{\hat{h}}_a = \{\hat{h}_a[1], \hat{h}_a[2], \ldots, \hat{h}_a[n]\}$ and $\underline{\hat{h}}_b = \{\hat{h}_b[1], \hat{h}_b[2], \ldots, \hat{h}_b[n]\}$, respectively, that are highly correlated, as in Figure 1. Although Eve can overhear the probe signals sent by each user, the signals received by her are completely different:

$$r_e^b(t_1) = s(t_1)h_{be}(t_1) + n_e(t_1) \quad (5)$$
$$r_e^a(t_2) = s(t_2)h_{ae}(t_2) + n_e(t_2), \quad (6)$$

where h_{be} and h_{ae} denote the channel between Bob & Eve and between Alice & Eve, respectively, and n_e is the noise added at Eve. If Eve is more than $\sim \lambda/2$ away from Alice and Bob, then h_{ae} and h_{be} are uncorrelated with $\mathbf{h}$ [23]. Therefore, despite possessing knowledge of the probe signal $s(t)$, Eve cannot use her received signals to compute meaningful estimates of the Alice-Bob channel, $\mathbf{h}$.

3.2 Converting the channel to bits

Alice and Bob must translate their respective sequences of channel estimates into identical bit-strings suitable for use as cryptographic keys, thus requiring: (1) *Suitably long–* Keys of length 128 to 512 bits are commonly used in symmetric encryption algorithms, and (2) *Statistically random–* The bits should not suffer from statistical defects that could be exploited by an attacker. The second requirement guarantees that the generated key has desirable security properties. That is, an N-bit key must provide N bits of uncertainty to an adversary who only knows the key generation algorithm..

We now briefly describe how to obtain bits from the channel estimates $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$, to provide the intuition behind our algorithm, while postponing a formal description to Section 4. The sequence of channel estimates $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$ are random variables drawn from an underlying probability distribution that characterizes the channel parameter $\mathbf{h}$. We assume, for the sake of discussion, that $h(t)$ is a Gaussian random variable and the underlying stochastic process $\mathbf{h}$ is a stationary Gaussian process. A Gaussian distribution for $\mathbf{h}$ may be obtained, for example, by taking $\mathbf{h}$ to be the magnitude of the in-phase component of a Rayleigh fading process between Alice and Bob [21]. We note that the assumption of a Gaussian distribution on $\mathbf{h}$ is for ease of discussion and our algorithm is equally valid in the general case.

Since the channel estimates computed by Alice and Bob are continuous random variables, it is necessary to quantize their estimates using a quantizer $Q(\cdot)$ to obtain bits. However, a straightforward quantization of the vectors $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$ is not sufficient because it does not guarantee that an identical sequence of bits will be generated at the two users. In our scheme, Alice and Bob use the channel statistics to determine scalars, q_+ and q_- that serve as reference levels for the quantizer $Q(\cdot)$ as follows:

$$Q(x) = \begin{cases} 1 & \text{if } x > q_+ \\ 0 & \text{if } x < q_- \end{cases} . \quad (7)$$

Alice parses through her channel estimates $\underline{\hat{h}}_a$ to determine the locations of *excursions* of her channel estimates above q_+ or below q_- that are of a duration $\geq m$ estimates, i.e., m successive channel estimates in $\underline{\hat{h}}_a$ are $> q_+$ or $< q_-$, where m is a protocol parameter. She sends Bob a message over the public channel containing the locations of k such excursions in the form of an array of indexes $L = \{l_1, l_2, \ldots, l_k\}$. Bob then checks his own sequence $\underline{\hat{h}}_b$ at the locations specified in L to determine whether it contains an excursion above q_+ or below q_- for a duration greater than or equal to '$m - 1$' samples, i.e. whether $\underline{\hat{h}}_a(l_i)$ is $> q_+$ or $< q_-$ for a duration that spans $m - 1$ or more estimates, for $i = 1, \ldots, k$. Bob identifies 'good' indexes by finding all index values l in L that produce such an excursion in $\underline{\hat{h}}_b$. He places these indexes into an array $\tilde{L}$ to be sent to Alice publicly. Indexes in L but not in $\tilde{L}$ are dropped from consideration by each party. The indexes in $\tilde{L}$ are used by each user to compute a sequence of bits by quantizing: $Q(\underline{\hat{h}}_a(\tilde{L}))$ and $Q(\underline{\hat{h}}_b(\tilde{L}))$. If the bit-vectors $Q(\underline{\hat{h}}_a(\tilde{L}))$ and $Q(\underline{\hat{h}}_b(\tilde{L}))$ are equal, then Alice and Bob succeed in generating $|\tilde{L}|$ identical bits. We show later that provided the levels q_+, q_- and the parameter m are properly chosen, the bits generated by the two users are identical with very high probability. A variation of the protocol that copes with spoofing is detailed in Section 4.1.

3.3 Design goals

An important quantity of interest will be the rate of generation of secret bits, expressed in secret-bits per second or 's-bits/sec'. Naturally, it is desirable that Alice and Bob achieve a high secret-bit rate. According to 802.1x recommendations, it is generally desirable for master keys to be refreshed at one hour intervals [24]. Using these examples and AES key sizes of 128 bits as a guideline, a conservative key rate of roughly 0.1 bits per second is needed, though it is desirable to achieve higher secrecy rates. However, we are especially wary of bit errors. If the sequence $Q(\underline{\hat{h}}_a(\tilde{L}))$ is different from $Q(\underline{\hat{h}}_b(\tilde{L}))$ even by a single bit, then the two bit-strings cannot be used as cryptographic keys and consequently the entire batch of bits must be discarded. Therefore, we would like the bit error probability p_e to be extremely low, so that the probability p_k that the keys generated by the two users do not match is acceptably small. For example, in order to have a key-mismatch probability of $p_k = 10^{-6}$, assuming keys of length 128 bits, we must target a bit-error probability of p_e where $p_k = 1 - (1 - p_e)^{128}$,

[3] 'Fast enough' here is in relation to the coherence time of the channel, which is inversely proportional to the maximum Doppler frequency f_d.

which gives $p_e \sim 10^{-8}$. A bit-error is defined as the event that Alice and Bob agree to use a certain index l_i contained in the list $\tilde{L}$ for generating a bit, but they end up generating different bits, i.e. $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$ both lie in excursions at the index l_i but the excursions are of opposite types.

The rate at which secret bits can be extracted from the channel is fundamentally limited by the rate of time-variation in the channel. We quantify this variation by the *maximum Doppler frequency*, f_d. A simple measure of the maximum Doppler frequency in a given wireless environment is given by $f_d = \frac{v}{\lambda}$, where v is a measure of the effects of user mobility and the dynamic environment around the users, expressed in meters/sec and λ is the wavelength of the carrier wave. In our case $\lambda = \frac{c}{f_0}$, where c is the speed of light. It can be seen that increasing the value m or the magnitudes of the quantizer boundaries q_+ & q_- would not only result in a lower rate, but also a lower probability of error. Intuitively, this is because larger magnitudes of q_+ & q_-, or a larger value of m makes it less likely that Alice's and Bob's channel estimates lie in opposite type of excursions, thereby reducing the error rate. However, both types of excursions also become less frequent, thereby decreasing the number of secret bits that can be generated per second. Thus, there is a tradeoff between rate and probability of error, and the parameters q_+, q_- and m provide convenient controls to select suitable operating points over this tradeoff. Beyond rate and robustness, we also require the bits to be random and free from statistical defects, as discussed in Section 5.3.

Finally, the correlated information obtained by Alice and Bob can be utilized to build a secret key in a number of different ways and it is important to make sure the method employed does not allow Eve to infer any useful information. An alternative bit extraction scheme is to have each user estimate a statistical measure of the channel (e.g. the mean signal-strength, or variance in the estimates) using $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$ respectively. If the channel is stochastically stationary, then their respective statistical measures would each converge to the true value with time. In this way, Alice and Bob will each possess knowledge about a numerical quantity, without having sent messages over the air containing this quantity. They could then quantize their estimates of the statistical measure to generate bits. However, the trouble with using a statistical measure is that knowledge of the locations of Alice and Bob and their environment may allow Eve to infer the statistics of the channel between them. Indeed, publicly available tools, such as the WISE ray-tracer [25], make it easy to predict the signal statistics at a receiver given the knowledge of the locations of the transmitter and receiver and the building's layout. Thus, it is important to recognize that using a statistical measure for key generation can be perilous. Our algorithm avoids statistical measures by relying on specific instantiations of the fading process.

4. LEVEL-CROSSING ALGORITHM

We now detail our level-crossing based key-extraction algorithm. It is assumed that when the algorithm is run, Alice and Bob have collected a sufficiently large number of channel estimates $\underline{\hat{h}_a}$ and $\underline{\hat{h}_b}$, by alternately probing the channel between themselves. Further, it is assumed that the vectors $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$ are of equal length and their j^{th} elements $\hat{h}_a(j)$ and $\hat{h}_b(j)$ correspond to successive probes sent by Bob and Alice respectively, for each $j = 1, \ldots, length(\underline{\hat{h}}_a)$. Algorithm 1 describes the procedure and consists of the following steps:

1. Alice parses the vector $\underline{\hat{h}}_a$ containing her channel estimates to find instances where m or more successive estimates lie in an excursion above q_+ or below q_-.

2. Alice selects a random subset of the excursions found in step 1 and for each selected excursion, she sends Bob the index of the channel estimate lying in the center of the excursion, as a list L. Therefore, if $\underline{\hat{h}}_a(i) > q_+$ or $< q_-$ for some $i = i_{start}, \ldots, i_{end}$, then she sends Bob the index $i_{center} = \lfloor \frac{i_{start}+i_{end}}{2} \rfloor$.

3. For each index from Alice, Bob checks whether his estimates $\underline{\hat{h}}_b$ contains *at least* $m-1$ channel estimates centered around that index in an excursion above q_+ or below q_-, i.e. whether $\underline{\hat{h}}_a > q_+$ or $< q_-$ for each index $\left\{l - \lfloor\frac{m-2}{2}\rfloor, \ldots, l + \lceil\frac{m-2}{2}\rceil\right\}$, for each $l \in L$.

4. For some of the indexes in L, Bob's channel estimates do not lie in either excursion. Bob makes a list $\tilde{L}$ of all indexes that lie in excursions and sends it to Alice.

5. Bob and Alice compute $Q(\underline{\hat{h}}_a)$ and $Q(\underline{\hat{h}}_b)$ respectively at each index in $\tilde{L}$, thus generating a sequence of bits.

Algorithm 1: The basic level crossing algorithm

```
Input  : $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$
Output : A cryptographic key $K_a = K_b$ at Alice and Bob
Alice:
for $i$ = 1 to $length(\underline{\hat{h}}_a) - m$ do
    if $Q(\hat{h}_a[i]) = Q(\hat{h}_a[i+1]) \ldots = Q(\hat{h}_a[i+m-1])$ then
        $i_{end} \leftarrow$ last index in excursion
        $L' \leftarrow [L' \quad ; \lfloor \frac{i+i_{end}}{2} \rfloor]$
        $i \leftarrow i_{end} + 1$
    else
        $i \leftarrow i + 1$
    end
end

$L$ = Random subset of $L'$
Alice sends $L$ to Bob on PUBLIC_CHANNEL .

Bob:
for $l \in L$ do
    if $Q\left(\hat{h}_b(l - \lfloor\frac{m-2}{2}\rfloor)\right) = \ldots = Q\left(\hat{h}_b(l + \lceil\frac{m-2}{2}\rceil)\right)$ then
        $\tilde{L} \leftarrow [\tilde{L}; \quad l]$
    end
end
$K_b = Q\left(\hat{h}_b(\tilde{L})\right)$
Bob sends $\tilde{L}$ to Alice on PUBLIC_CHANNEL.

Alice:
$K_a = Q(\hat{h}_a(\tilde{L}))$
```

Since Eve's observations do not provide her with any useful information about $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$, the messages L and $\tilde{L}$ do not provide her any useful information either. This is because they contain time indexes only whereas the generated bits depend upon the values of the channel estimates at those indexes. Further, the selection of a random subset from the set of eligible excursions, guarantees that Eve cannot use L and $\tilde{L}$ to infer the values of the channel estimates of Alice or Bob at those time indexes.

4.1 Preventing a Spoofing Attack

Since Alice and Bob do not share an authenticated channel, Eve can impersonate Alice in Step 2, or Bob in Step 4 above. Such an attack would allow Eve to insert her own 'fake' L or $\tilde{L}$ messages, thus spoofing a legitimate user and disrupting the protocol without revealing her presence. Therefore we require a form of *data-origin authentication*, that assures each user that the L or $\tilde{L}$ message has originated at the legitimate transmitter. Our protocol can be made to detect the adversary in each of the two cases above. We first focus on Eve inserting a fake L-message. Since Eve has no information about the locations of channel excursions apart from L, she can only make random guesses about which indexes to place into a fake L-message to Bob (apart from the ones Eve learns from L). If Eve inserts a significant number of random guesses into a fake L-message, Bob can detect her presence by computing the proportion of indexes in L that lead to excursions in $\underline{\hat{h}}_b$. Since Eve can only make random guesses, this quantity would be much lower than one resulting from a legitimate L-message from Alice. For each guess, she has a very low probability of choosing an index that lies in an excursion spanning $(m-1)$ or more estimates at Bob. Of these, the indexes that do not lie in an excursion in $\underline{\hat{h}}_b$ are discarded by Bob while those that do, are considered eligible for quantization and placed into the $\tilde{L}$-message sent to Alice. Thus, an unsuccessful guess provides no benefit to Eve, while a successful guess, albeit improbable, causes $\tilde{L}$ to contain an index that was not present in L, thereby alerting Alice. Thus, Eve must also modify $\tilde{L}$ by deleting this index before it reaches Alice. Our protocol can be made to resist modification of the $\tilde{L}$-message using a *message authentication code* (MAC), by the following *additional* steps:

Algorithm 2: Modified algorithm incorporating data-origin authentication and resistance to an active attack.

```
Input   : $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$
Output  : A cryptographic key $\bar{K}_a = \bar{K}_b$ at Alice and Bob
Alice:
for $i$ = 1 to length($\underline{\hat{h}}_a$) $- m$ do
    if $Q(\hat{h}_a[i]) = Q(\hat{h}_a[i+1]) = \ldots = Q(\hat{h}_a[i+m-1])$ then
        $i_{end} \leftarrow$ last index in excursion
        $L' \leftarrow [L' \quad ; \lfloor \frac{i+i_{end}}{2} \rfloor]$
        $i \leftarrow i_{end} + 1$
    else
        $i \leftarrow i + 1$
    end
end
$L$ = Random subset of $L'$
Alice sends $L$ to Bob on PUBLIC_CHANNEL.

Bob:
for $l \in L$ do
    if $Q\left(\hat{h}_b(l - \lfloor\frac{m-2}{2}\rfloor)\right) = \ldots = Q\left(\hat{h}_b(l + \lceil\frac{m-2}{2}\rceil)\right)$ then
        $\tilde{L} \leftarrow [\tilde{L}; \quad l]$
    end
end
if $\left\{\frac{|\tilde{L}|}{|L|} < 0.5 + \epsilon\right\}$ then
    DECLARE ACTIVE ATTACK
else
    $K_b = Q\left(\hat{h}_b(\tilde{L})\right)$
    $K_{au} = K_b(1, \ldots, N_{au})$
    $\bar{K}_b = K_b(N_{au}+1, \ldots, N)$
    $Package = \left\{\tilde{L}, MAC\left(K_{au}, \tilde{L}\right)\right\}$
    Bob sends $Package$ to Alice on PUBLIC_CHANNEL.
end

Alice:
$K_a = Q\left(\hat{h}_a(\tilde{L})\right)$
$K_{au} = K_a(1, \ldots, N_{au})$
$\bar{K}_a = K_a(N_{au}+1, \ldots, N)$
if MAC validation using $K_{au}$ fails then
    DECLARE ACTIVE ATTACK
end
```

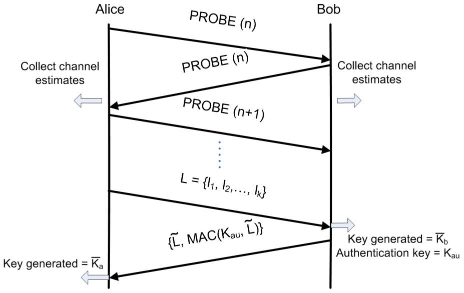


Figure 2: Timing diagram for the key-extraction protocol.

1. To make sure the L-message received is from Alice, Bob computes the fraction of indexes in L where $\underline{\hat{h}}_b$ lies in an excursion spanning $(m-1)$ or more estimates. If this fraction is less than $\frac{1}{2}+\epsilon$, for some fixed $0 < \epsilon < \frac{1}{2}$, Bob concludes that the message wasn't sent by Alice, implying an adversary has injected a fake L-message.

2. If the check above passes, Bob replies to Alice with a message $\tilde{L}$ containing those indexes in L at which $\underline{\hat{h}}_b$ lies in an excursion. Bob computes $K_b = Q(\underline{\hat{h}}_b(\tilde{L}))$ to obtain N bits. The first N_{au} bits are used as an authentication key to compute a message authentication code (MAC) of $\tilde{L}$. The remaining $N - N_{au}$ bits are kept as the extracted secret key. The overall message sent by Bob is $\{\tilde{L}, MAC(K_{au}, \tilde{L})\}$.

Upon receiving this message from Bob, Alice uses $\tilde{L}$ to form the sequence of bits $K_a = Q(\hat{h}_a(\tilde{L}))$. She uses the first N_{au} bits of K_a as the authentication key $K_{au} = K_a(1, \ldots, N_{au})$, and using K_{au} she verifies the MAC to confirm that the package was indeed sent by Bob. Since Eve does not know the bits in K_{au} generated by Bob, she cannot modify the $\tilde{L}$-message without failing the MAC verification at Alice.

Even without an authenticated channel, Alice and Bob can successfully establish a common secret key despite an active adversary, provided there are no bit errors. This explains why we insist on a very low probability of error in Section 3.3. Further, the reduction in the secret-bit rate due the to N_{au} bit is negligible because they are a one-time expense enabling Alice and Bob to bootstrap data-origin authentication. A modified algorithm that incorporates the above ideas is presented as Algorithm 2 (see Figure 2). Another active attack involves Eve impersonating Alice or Bob during the channel-probing stage, i.e. Eve may begin sending probes to Bob pretending to be Alice or vice-versa. Such an attack can be detected using a hypothesis testing approach on the recent history probes received at each legitimate user, and this has been extensively studied in [14,15]. The technique relies on the insight that given a sufficiently fast probing rate, successive probes received by a user are most likely to differ by a small amount. We provide further discussion related to the security of our scheme in Section 7.

5. PERFORMANCE EVALUATION

The central quantities of interest in our protocol are the rate of generation of secret bits, the probability of error and the randomness of the generated bits. The controls available to us are the parameters: q_+, q_-, m and the rate at which Alice and Bob probe the channel between themselves, f_s. We assume the channel is not under our control, and as explained in Section 3.3, the rate at which the channel varies can be represented by the maximum Doppler frequency, f_d. The typical Doppler frequency for indoor wireless environments at the carrier frequency of 2.4 GHz is $f_d = \frac{v}{\lambda} \sim \frac{2.4\times 10^9}{3\times 10^8} = 8$ Hz, assuming a velocity v of 1 m/s. We thus expect typical Doppler frequencies in indoor environments in the 2.4 GHz range to be roughly 10 Hz and 20 Hz in the 5 GHz range. For automobile scenarios, we can expect a Doppler of ~ 200 Hz in the 2.4 GHz range.

5.1 Probability of error

The probability of error, p_e is critical to our protocol. In order to achieve a robust key-mismatch probability p_k, the bit-error probability p_e must be much lower than p_k. A bit-error probability of $p_e = 10^{-7} \sim 10^{-8}$ is desirable for keys of length $N = 128$ bits. We have explained in Section 3.3 that there is a fundamental trade-off in the selection of parameters m, q_+ and q_- that affects the rate and probability of error in opposing ways. The probability of bit-error, p_e is the probability that a single bit generated by Alice and Bob is different at the two users. The symmetry of the distribution of $\mathbf{h}$ allows us to consider just one type of bit error in computing p_e. Consider the probability that Bob generates the bit "0" at an index given that Alice has chosen this index but she has generated the bit "1". As per our Gaussian assumption on the parameter $\mathbf{h}$ and estimates $\hat{h}_a$ and $\hat{h}_b$, this probability can be expanded as

$$P(B=0|A=1) = \frac{P(B=0, A=1)}{p(A=1)} = \tag{8}$$

$$\frac{\underbrace{\int_{q_+}^{\infty}\int_{-\infty}^{q_-}\cdots\int_{q_+}^{\infty}}_{(2m-1)\ terms} \frac{(2\pi)^{(1-2m)/2}}{|K_{2m-1}|^{1/2}} \exp\left\{-\frac{1}{2}x^T K_{2m-1}^{-1} x\right\} d^{(2m-1)}x}{\underbrace{\int_{q_+}^{\infty}\cdots\int_{q_+}^{\infty}}_{(m)\ terms} \frac{(2\pi)^{-m/2}}{|K_m|^{1/2}} \exp\left\{-\frac{1}{2}x^T K_m^{-1} x\right\} d^{(m)}x},$$

where K_m is the covariance matrix of m successive Gaussian channel estimates of Alice and K_{2m-1} is the covariance matrix of the Gaussian vector $(\hat{h}_a[1], \hat{h}_b[1], \hat{h}_a[2], \ldots, \hat{h}_b[m-1], \hat{h}_a[m])$ formed by the combining m channel estimates of Alice and the $m-1$ estimates of Bob in chronological order. The numerator in (8) is the probability that of $2m-1$ successive channel estimates (m belonging to Alice, and $m-1$ for Bob), all m of Alice's estimates lie in an excursion above q_+ while all $m-1$ of Bob's estimates lie in an excursion below q_-. The denominator is simply the probability that all of Alice's m estimates lie in an excursion above q_+. We compute these probabilities for various values of m and present the results of the probability of error computations in Figure 3. The results confirm that a larger value of m will result in a lower probability of error, as a larger m makes it less likely that Alice's and Bob's estimates lie in opposite types of excursions. Note that if either user's estimates do not lie in an excursion at a given index, a bit error is avoided because that index is discarded by both users.

5.2 Secret-bit rate

The correct way to address the tradeoff between probability of error and rate of generation of secret bits is to upper bound the acceptable probability of error and then attempt to derive the greatest possible rate. How many s-bits/second can we expect to derive from a time-varying channel? An approximate analysis can be done using the level-crossing rate for a Rayleigh fading process, given by $LCR = \sqrt{2\pi} f_d \rho e^{-\rho^2}$ [21], where f_d is the maximum Doppler frequency and ρ is the threshold level, normalized to the root mean square signal level. Setting $\rho = 1$, gives $LCR \sim f_d$.

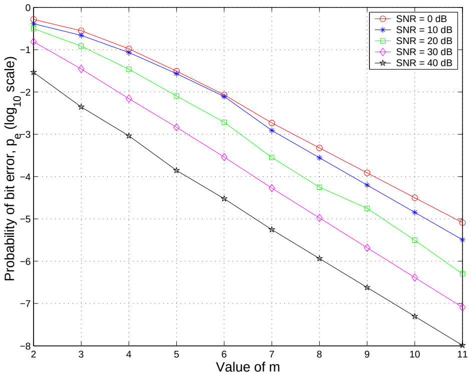


Figure 3: Probability of bit error p_e for various values of m at different SNR levels ($q_\pm = mean \pm 0.8\sigma$)

The above calculation tells us that we cannot expect to obtain more s-bits per second than the order of f_d. In practice, the rate of s-bits/sec depends also on the channel probing rate f_s, i.e. how fast Alice and Bob are able to send each other probe signals. In Figure 4 (a) and (b), we plot the rate in s-bits/sec as a function of the channel probing rate for a wireless channel with maximum Doppler frequencies of $f_d = 10$ Hz and $f_d = 100$ Hz respectively. As expected, the number of s-bits the channel yields increases with the probing rate, but saturates at a value on the order of f_d. More precisely, the number of s-bits/sec is the number of s-bits per observation times the probing rate. Therefore

$$R_k = H(bins) \times p(A=B) \times \frac{f_s}{m} \tag{9}$$

$$= 2\frac{f_s}{m} \times p(A=1, B=1) \tag{10}$$

$$=2\frac{f_s}{m}.\underbrace{\int_{q_+}^{\infty}\cdots\int_{q_+}^{\infty}}_{(2m-1)\ terms} \frac{(2\pi)^{\frac{1-2m}{2}}}{|K_{2m-1}|^{1/2}} e^{\left\{-\frac{1}{2}x^T K_{2m-1}^{-1} x\right\}} d^{2m-1}x, \tag{11}$$

where $H(bins)$ is the entropy of the random variable that determines which bin ($> q_+$ or $< q_-$) of the quantizer the observation lies in, which in our case equals 1 assuming that the two bins are equally likely (The levels q_+ and q_- are chosen so as to maintain equal probabilities for the two bins). The probing rate f_s is normalized by a factor of m because a single 'observation' in our algorithm is a sequence of m channel estimates. The expression in (11) is reminiscent of the probability of error expression in (8) and has been evaluated in Figure 4. Figure 4 confirms the intuition that the secret bit rate must fall with increasing m, since the longer duration excursions required by a larger value of m are less frequent. In Figure 5 (a), we investigate how the secret-bit rate R_k varies with the maximum Doppler frequency f_d, i.e. versus the channel time-variation. We found that for a fixed channel probing rate (in this case, $f_s = 4000$ probes/sec), increasing f_d results in a greater rate but only up to a point, after which the secret-bit rate begins to fall. Thus, 'running faster' does not always help unless we can increase the probing rate f_s proportionally. This suggests that not only does each channel have an optimal minimum probing rate for de-

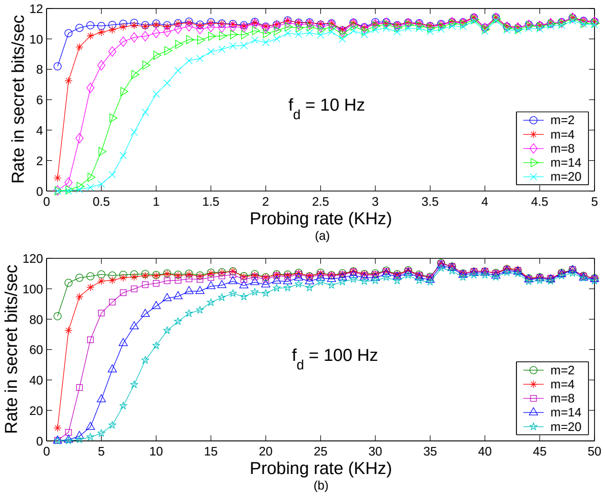
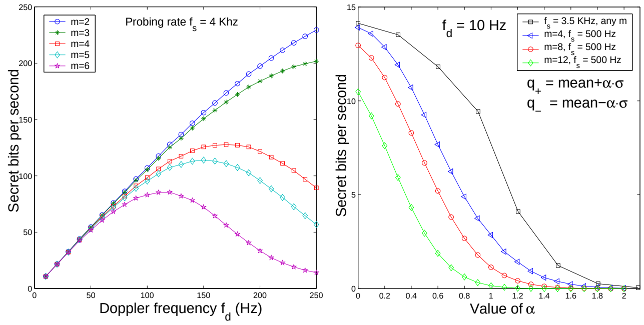


Figure 4: Rate in secret bits per second for various values of m, against probing rate for a channel with Doppler frequency (a) $f_d = 10$ Hz and (b) $f_d = 100$ Hz ($q_{\pm} = mean \pm 0.8\sigma$)



(a) (b)

Figure 5: (a) Secret-bit rate for varying Doppler f_d and fixed f_s for various values of m (b) Rate as a function of function of quantizer levels q_+ & q_- parametrized by α.

riving the best possible secret-bit rate, but each probing rate also corresponds to a most 'useful' maximum Doppler frequency. Figure 5(b) shows the expected decrease in rate as the quantizer levels q_+ and q_- are increased in magnitude. In this figure, α denotes the number of standard deviations from the mean at which the quantizer levels are placed.

5.3 Randomness of generated bits

Guaranteeing that the generated bits are random is crucial because they are intended for use as a cryptographic key. Since we have assumed the adversary possesses complete knowledge of our algorithm, any non-random behavior in the bit sequences can be exploited by the adversary to reduce the time-complexity of cracking the key. For example, if the algorithm is known to produce a greater proportion of '1's than '0's, then the effective search space for the adversary would be reduced. Consequently, a variety of statistical tests have been devised to test for various defects [26].

In evaluating the randomness of bit sequences generated by our algorithm, we focus on Maurer's universal statistical test [27], a widely accepted benchmark for testing randomness. The test statistic relates closely to the per-bit entropy of the sequence, and thus measures the actual cryptographic significance of a defect as related to the running time of an adversary's optimal key-search strategy [27].

Test	P-value
Maurer's Test	0.8913
Monobit frequency	0.9910
Runs Test	0.1012
Approx. entropy	0.8721
Random excursions	0.5829
Lempel Ziv	1.0000

Table 2: Results from randomness tests on bit sequences (10^8 bits) produced by our algorithm for $f_d = 10$ Hz, $f_s = 30$ Hz, $m = 5$ and $q_+, q_- = mean \pm 0.2\sigma$. In each test, a p-value > 0.01 indicates the sequence is random.

Additionally, we ran a few other tests using the NIST public-domain test suite. We refer the interested reader to [28] for a description of these tests and the definitions of $p - value$ for each test. The results for these are summarized in Table 2. Subsequent runs produced comparable results and thus support the conclusion that our algorithm provides random bits. In particular, Maurer's test showed the average entropy of our bit-sequences is very close to the value expected for a truly random sequence. This can be possible only if successive bits are almost independent, which in turn requires that they must be separated in time by at least a 'coherence time' interval. Since the coherence time of a channel is inversely proportional to the Doppler frequency, extracting bits from a channel's level-crossings at a rate significantly greater than f_d cannot possibly produce random bits. We observed in Section 5.2 that the rate at which our algorithm generates secret bits is bounded from above by approximately the maximum Doppler f_d. Finally, we note that the selection of a random subset of excursions by Alice effectively allows her some control on selecting the final key generated. Thus, even if a particular run happens to produce excursions at Alice containing a statistical defect in the resulting bit sequence, she can fix the defect to some extent by suitably choosing L from among eligible excursions.

6. VALIDATION USING 802.11A

We now describe our experimental validation efforts for typical indoor environments. Our experiments were divided in two parts. In the first study, we delved into the structure of an 802.11 packet to access the preamble sequence [29] in the received signal to compute a 64-point *channel impulse response* (CIR) that showed one or more resolvable dominant paths as separate peaks. We used the magnitude of the tallest peak in the CIR (the dominant multipath) as the channel our parameter of interest. To access signal information at the sample level, we used an 802.11 development platform with FPGA-based customized logic added for processing CIR. Our results showed that our algorithm works very well for both static and mobile scenarios, producing error-free secret bits at rates ~ 1 s-bits/sec in the tested indoor environments.

We then sought to determine whether off-the-shelf 802.11 hardware could achieve comparable results. Thus, for the second study, we used RSSI measurements reported in the Prism headers of 802.11 packets on commercially available 802.11a cards, with Alice configured as an access point (`AP` mode) and Bob as a client (`station` mode), and a third user configured to listen (`station` mode) on transmissions from both legitimate users.

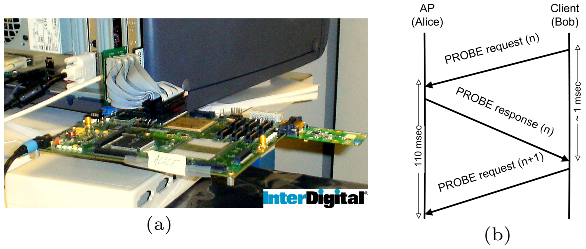


Figure 6: (a) Our experimental platform - a development board for a commercial 802.11a/b/g modem IP, to which we added custom logic to process CIR information. (b) Timing diagram for collecting CIR information using PROBE packets

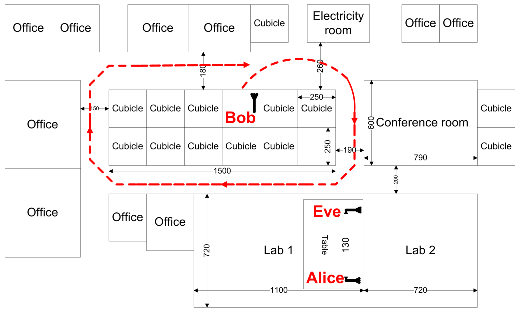


Figure 7: A layout of the experimental setup for the CIR method (distances in cm)

6.1 CIR method using 802.11a

Experiment setup: Our experimental platform (Figure 6(a)) consisted of an 802.11 development board with commercial 802.11a/b/g modem IP, to which we added custom logic to extract the channel impulse response from received packets. This allowed us to pull out received signal information at a level not normally accessible using commodity 802.11 hardware and drivers. Two such boards were set up as Alice and Bob, while a third board was configured to be Eve. Alice was configured to be an access point (`AP`), and Bob was configured to be a client (`station`). The experiment involved Bob sending `PROBE request` messages to Alice, who then replied with a `PROBE response` (Figure 6(b)). Limitations of our development boards allowed us to have Eve listen on either Alice or Bob, but not both. In the results presented here, Eve has been configured to listen in on Alice. In the first experiment, Alice and Eve were placed in a laboratory, while Bob was placed in an office cubicle outside the lab, see Figure 7. In the second experiment, Alice and Eve remained in the same positions while Bob circled the cubicle area along the trajectory in Figure 7 in a cart on wheels. Figure 8 shows a 64-point CIR obtained from a single 802.11a `PROBE request` packet received at Alice, along with the corresponding CIR computed from the `PROBE response` packet received by Bob in reply. Also shown is the CIR as computed by Eve, using the overheard `PROBE response` packet from Alice. For our algorithm, we use only the magnitude of the main peak in the CIR.

Figure 9 shows the traces of the CIR's main peak's magnitude at Alice and Bob for our first experiment. While our experiment ran for ~ 22 minutes, in the interest of space and clarity we show 700 CIRs collected over a duration of ~ 77 seconds. The traces show significant changes in average signal power, ostensibly due to time-variations in the wireless environment between Alice and Bob (see Figure 7). If each user simply uses this data as input to the level-crossing bit-extraction algorithm, the generated key has long strings of 1s and 0s (see Figure 9). This is because we are attempting to include the effect of *shadow fading* [21] (also called large-scale fading) that produces large but slow swings in the average signal power into the key generation algorithm. In other words, the channel in Figure 9 is not stationary. Each user locally computes q_+ and q_- as: $q_+^u = mean(\underline{\hat{h}}_u) + \alpha \cdot \sigma(\underline{\hat{h}}_u)$ and $q_-^u = mean(\underline{\hat{h}}_u) - \alpha \cdot \sigma(\underline{\hat{h}}_u)$, where u can be Alice or Bob, $\underline{\hat{h}}_u$ is the set of magnitudes of the CIR's main peak collected by user u, and $\sigma(\underline{\hat{h}}_u)$ represents the standard deviation of $\underline{\hat{h}}_u$. The factor α can be selected to vary the quantizer levels. We chose $\alpha = \frac{1}{8}$ for the CIR-method. The effect of the underlying shadow fading contained in the collected data can be removed by subtracting a moving average of each trace from the original trace. This leaves only the small scale fading that we wish to use in our algorithm. The result is shown in Figure 10. In this way, not only do we do away with the problem of long strings of 1s and 0s, we also prevent the average signal power from affecting our key generation process. Using the small scale fading traces, our algorithm generates $N = 125$ s-bits in 110 seconds ($m = 4$), yielding a key rate of about 1.13 s-bits/sec.

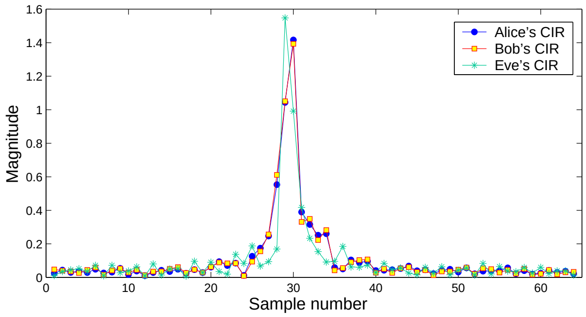


Figure 8: The 64-point CIR from a single 802.11 packet. For our key-extraction algorithm, we use the magnitude of the main peak as the channel parameter of interest.

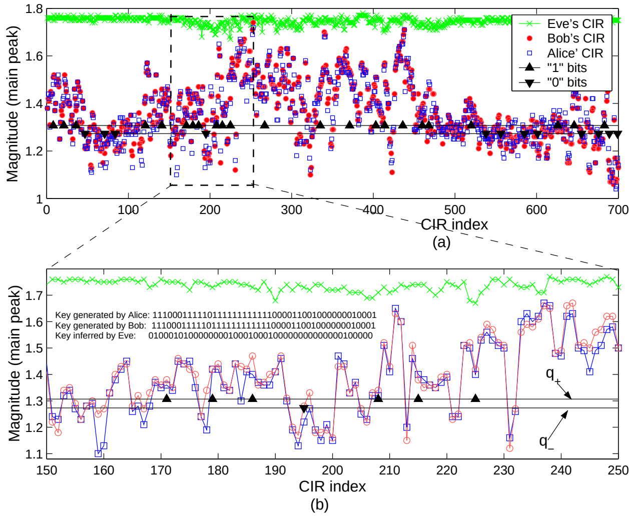


Figure 9: (a) Traces of Alice, Bob and Eve. Variation in avg. signal power produces longs strings of 1s and 0s. (b) A magnified portion of the traces.

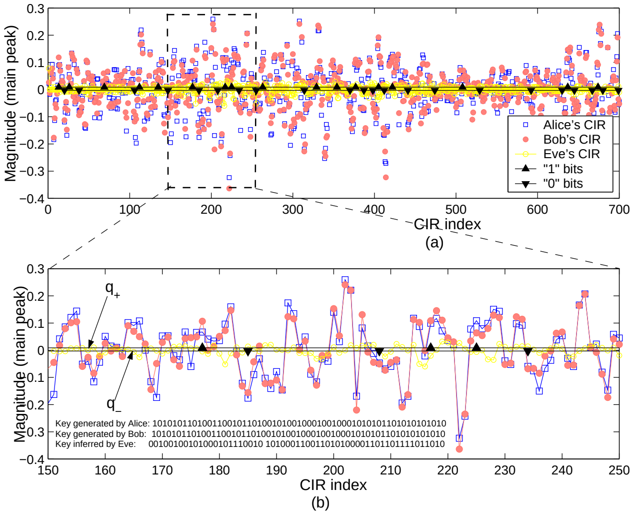


Figure 10: (a) Traces of Alice and Bob after subtracting average signal power. Using $m = 5$, $N = 59$ bits were generated in 110 seconds ($R_k = 0.54$ s-bits/sec) while $m = 4$ gives $N = 125$ bits ($R_k = 1.13$ s-bits/sec.) with no errors in each case. (b) A magnified portion of (a)

Contrasting Eve's attempts: Figures 9 shows a trace of Eve's CIR peak as overheard from Alice along with Alice's and Bob's traces. Figure 10 shows the bits that Eve would generate if she carried through with the key-generation procedure. The mutual information [8] (M.I.) between Eve's data and Bob's data is a useful measure of the information learned by Eve about Bob's measurements $\underline{\hat{h}}_b$ and can be compared to the mutual information between Alice's and Bob's estimates $\underline{\hat{h}}_a$ and $\underline{\hat{h}}_b$. Table 3 gives these mutual information values computed using the method in [30]. As a consequence of the data processing inequality [8], any processing of the received signal by Eve would only reduce her information about the Alice-Bob channel, and therefore, the M.I. values in Table 3 provide upper bounds on the information about the Alice-Bob channel leaked out to Eve. The results from our second experiment with a moving Bob are very similar to the ones shown for the first experiment, although with fewer bits produced. Due to space limits, we do not present plots for the mobile experiment but instead summarize our results in Table 3. It is notable that in the static case, the M.I. between Eve and Bob is orders of magnitude smaller than that between Alice and Bob and very close to zero, indicating that Eve is unable to derive any significant information about the Alice-Bob channel. Further, the M.I. between Eve and Bob is lower in the mobile case compared to the static case, indicating that mobility actually helps strengthen the secrecy of generated keys.

6.2 Coarse measurements using RSSI

Experiment setup: The setup consisted of three off-the-shelf 802.11 radios. Alice was configured in `AP` mode along with a virtual monitor interface to capture received packets. Bob was a client, consisting of a laptop with a 802.11a card in `station` mode, along with virtual monitor for capturing received packets. Eve was a third 802.11a node, identical in configuration to Bob, but capable of receiving packets from both Alice and Bob. In our experiment, Alice was stationary, while Bob and Eve moved along fixed trajectories. Atheros WiFi cards based on the 5212 chipset were used at each end along with the Madwifi driver for Linux. The experiments were done in the 5.26 GHz channel. The AP-station configuration ensured that MAC-layer clocks at the two nodes were synchronized. Figure 11 (b) shows the layout of the office building along with the location of the fixed AP and path followed by the mobile client. ICMP `PING` packets were sent from the AP to the client at a rate of 20 packets per second. Each `PING request` packet received at the client generates a MAC-layer acknowledgment packet sent back to the AP, followed by a `PING response` packet. Upon receiving the `PING response` packet, the AP similarly replies with a MAC-layer ACK packet. Figure 11 (a) shows the sequence in which these packets are sent. A `tcpdump` application running on both the AP and the client recorded and time-stamped all packets received on the monitor interface of each user. The experiment consisted of sending 8,000 packets from Alice to Bob. The `tcpdump` traces at each end were filtered using the MAC address to keep only the four types of packets described above. Further, RSSI and MAC-timestamps were pulled out of each packet to generate a $(timestamp, RSSI)$ trace.

Since we did not have index numbers with which to reference RSSI values, Alice sends MAC-timestamps in the mes-

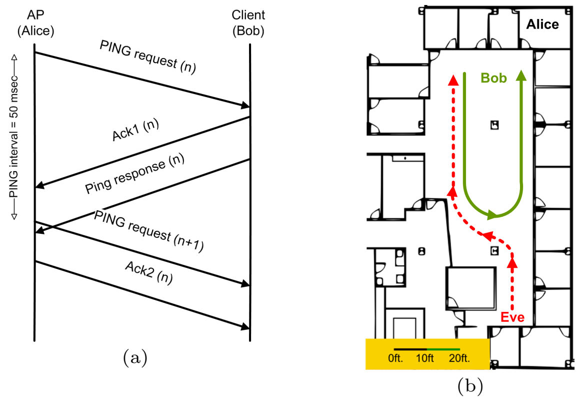


Figure 11: (a) Timing diagram for collecting RSSI information using PING packets in the RSSI-method. (b) Experimental Layout for RSSI-based method showing trajectories of Bob and Eve, while Alice (the AP) was kept stationary.

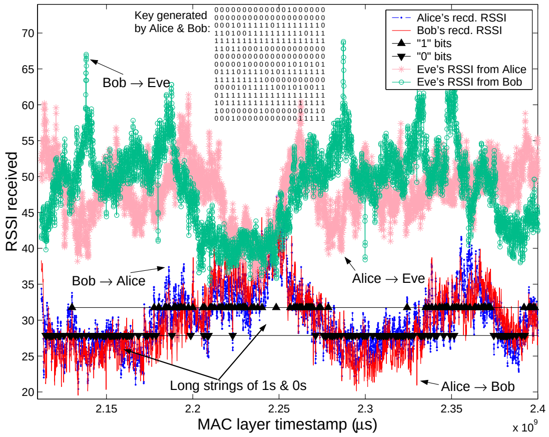


Figure 12: RSSI traces of Alice and Bob and bits generated. This plot includes the effect of shadow fading.

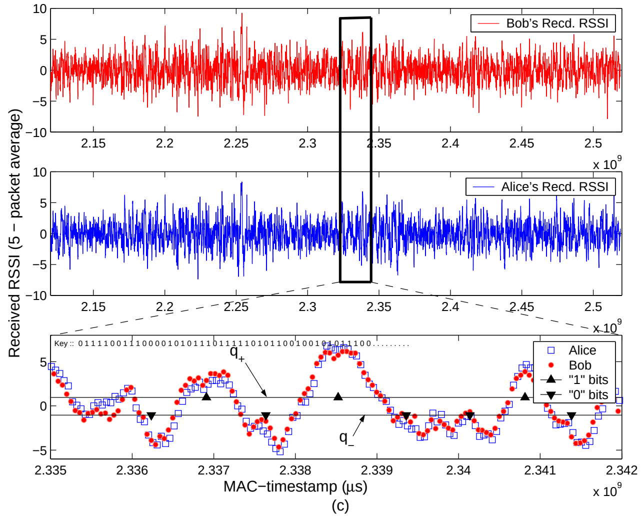


Figure 13: RSSI traces of Alice & Bob after subtracting windowed mean. We get 511 bits in 392 sec using $m = 4$ ($R_k = 1.3$ s-bits/sec.)

CIR-based method

Value of m used	4
Choice of q_+, q_-	mean $\pm 0.125\sigma$
Duration of experiments	1326 sec ($\sim$ 22 min.)
Inter-probe duration	110 msec.
Static case:	
Average secret-bit rate	1.28 s-bits/sec.
I(Alice; Bob)	3.294 *bits*
I(Bob; Eve)	0.0468 *bits*
Mobile case:	
Average secret-bit rate	1.17 s-bits/sec.
I(Alice; Bob)	1.218 *bits*
I(Bob; Eve)	0.000 *bits*

RSSI-based method

Value of m used	4
Choice of q_+, q_-	mean $\pm 0.5\sigma$
Average secret-bit rate	1.3 s-bits/sec
Inter-probe duration	50 msec.
Duration of experiment	400 sec.
I(Alice; Bob)	0.78 *bits*
I(Alice; Eve)	0.00 *bits*
I(Bob; Eve)	0.07 *bits*

Table 3: Summary of experimental results. $I(u_1; u_2)$ denotes the mutual information (M.I.) between the measurements of users u_1 and u_2.

sage L (see Algorithm 1 in Section 4). For each timestamp from Alice, Bob finds the timestamp in his own trace that is closest to the one sent by Alice and uses it to check for excursions above q_+ or below q_- as in Algorithm 1.

The *RSSI* field in the Prism header of received 802.11 packets reports RSSI as integers, thereby providing only coarse channel information. Moreover, the 802.11 cards at Alice and Bob may not be relatively calibrated and thus may report different values of RSSI. We found in our experiments that although lacking calibration, the temporal *variations* in RSSI are matched in Alice's and Bob's traces. This problem was solved by subtracting out a moving average of the trace to remove the effects of slowly varying average signal power, as in the CIR method. Figure 12 shows the raw RSSI traces collected by Alice and Bob plotted against their received MAC-timestamps. As in the CIR-method, the traces exhibit strong variations in average signal power. We average out the large-scale variations and keep only the small scale fading effect. The result is shown in Figure 13. Our algorithm produces secret bits at a rate of almost 1.3 s-bits/sec using $m = 4$, where q_+ and q_- were computed independently by each user as in Section 6.1 with $\alpha = \frac{1}{2}$.

Contrasting Eve's attempts: We plot the RSSI traces captured by Eve for both Alice's and Bob's signal in Figure 12. The traces from Alice and Bob after considering only variations about a moving average, are shown in Figure 13. Even with coarse RSSI measurements that represent the average received signal power per-packet over the entire 802.11 channel bandwidth, Alice and Bob can exploit reciprocity of their channel to successfully generate secret bits at a fairly good rate. We compute the pair-wise M.I. between the traces of Eve, Alice and Bob in Table 3. As in the CIR-method, we find that Eve gets almost no information about the Alice-Bob channel.

7. DISCUSSION

The natural decorrelative properties of fading provides our scheme security against eavesdroppers. We confirmed this through our system implementation. Standard randomness tests indicate that our algorithm is resilient to an eavesdropper exploiting randomness defects. However, it is worth noting that key rates significantly greater than the maximum Doppler frequency cannot result in truly random bits. Thus we recommend conservatively setting the probing rates relative to the dynamics of the fading environment. Beyond a passive adversary, we have addressed the threat of an active adversary impersonating Alice or Bob. Coping with spoofing of probes can be dealt with using techniques similar to [15]. We have addressed spoofing of messages following probing by providing a modified algorithm that uses some of the shared secret bits for data-origin authentication. Thus, Eve cannot thwart the key-generation process by impersonating either legitimate user without getting detected.

A further concern common to all key establishment schemes is the man-in-the-middle attack. A man-in-the-middle attack against our algorithm is only possible if Alice and Bob cannot hear each other's probes (e.g. they are not within radio range, or Eve talks to Alice and Bob separately), otherwise Eve's attack causes discrepancies that are easily detectable by Alice and Bob. If Alice and Bob do fall victim to a man-in-the-middle attack, this can be detected by the following identity-based authentication mechanism: Alice asks Bob to send her the keyed hash of the answer to a specific question using their (supposed) shared key as an input to a cryptographic hash function. If Eve relays this question to Bob, then Bob's answer will be useless to Eve (assuming only Alice and Bob know the answer to the question). We note this method requires that Alice and Bob share some secret information known only to them. This is necessary as each user must authenticate the *identity* of the other in order to prevent a man-in-the-middle attack, and is necessary even for classical key establishment schemes like Diffie-Hellman. Finally, the astute reader might inquire whether varying levels of interference at different locations in the environment would affect our key generation process. We have provided fundamental tradeoffs relating signal-to-interference levels to quantizer parameter selection for an isotropic noise background. However, by conservatively selecting protocol parameters (e.g. selecting a larger value of

m (see Figure 3)), we achieve improved robustness in the key generation process at the cost of lowering the rate.

8. CONCLUSIONS

In this paper, we proposed a protocol that exploits the reciprocity of the transfer function of the wireless multipath channel to establish a common cryptographic key between two communicating entities. Our protocol obtains a security advantage from the fact that the channel response decorrelates rapidly with distance from each communicator, implying that there is strong protection against a passive eavesdropper as well as an active adversary attempting a spoofing attack. The performance of our scheme was evaluated and important insights relating the probing rate, quantizer parameters and the resulting secret key rate were provided.

We also presented the results of a thorough effort to experimentally validate the utility of the wireless channel for secret key generation. First, we constructed a system to extract channel impulse responses on a customized 802.11 development platform, where we used the 802.11a preamble to compute channel impulse responses on a per-packet basis. Second, we used off-the-shelf 802.11a cards for collecting coarse RSSI measurements. In both cases, our algorithm generated secret bits at a useful rate without any errors. We showed that an eavesdropper shares minuscule mutual information with legitimate communicators, thereby supporting security against eavesdroppers. Our work demonstrates that the multipath information that is inherent in any wireless system (and is normally discarded after physical layer processing), can successfully support key establishment. More importantly, we showed that although this capability is possible with custom architectures, it can be achieved using off-the-shelf radio platforms, and thus could have immediate impact on the security of commodity wireless systems. Looking beyond our fundamental observations and feasibility studies, we note that our algorithm naturally applies to emerging wireless systems that use MIMO or OFDM to enhance data rates since the associated multiple uncorrelated channels between two users would lead to a proportional increase in the secret-bit extraction rate.

Acknowledgements: The authors would like to thank Yogendra Shah for discussions, and NSF and DARPA (CNS-0626439 and W31P4Q-07-1-0002) for support.

9. REFERENCES

[1] *Method and system for deriving an excryption key using joint randomness not shared by others.* InterDigital Communications Corporation, US Patent Application ITC-2-1135.01.WO, 2006.

[2] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 733–742, 1993.

[3] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography – Part I: Secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, 1993.

[4] J. Cardinal and G. V. Assche, "Construction of a shared secret key using continuous variables," *Info. Theory Workshop*, 2003.

[5] G. Brassard and L. Salvail, "Secret key reconciliation by public discussion," *Advances in Crytology Proc. - Eurocrypt '93, Lecture Notes in Computer Science*, vol. 765, pp. 410–423, 1994.

[6] C. Ye, A. Reznik, and Y. Shah, "Extracting secrecy from jointly Gaussian random variables," in *Proceedings of IEEE Int. Symp on Info. Theory*, Jul 2006, pp. 2593 – 2597.

[7] C. Cachin and U. M. Maurer, "Linking information reconciliation and privacy amplification," *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, vol. 10, no. 2, pp. 97–110, Spring 1997.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* John Wiley, 1991.

[9] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion," *SIAM J. Comput.*, vol. 17, no. 2, pp. 210–229, 1988.

[10] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, and C. G. Peterson, "Fast, efficient error reconciliation for quantum cryptography," *Phys. Rev. A*, vol. 67, pp. 052303.1-052303.8, 2003.

[11] G. V. Assche, *Quantum Cryptography and Secret Key Distillation.* Cambridge University Press, 2006.

[12] U. Maurer and S. Wolf, "Secret key agreement over a non-authenticated channel -Part II: The simulatability condition," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 832–838, Apr. 2003.

[13] Z. Li, W. Xu, R. Miller, and W. Trappe, "Securing wireless systems via lower layer enforcements," in *WiSe '06: Proceedings of the 5th ACM workshop on Wireless security*, 2006, pp. 33–42.

[14] N. Patwari and S. K. Kasera, "Robust location distinction using temporal link signatures," in *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, 2007, pp. 111–122.

[15] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Fingerprints in the ether: Using the physical layer for wireless authentication," in *Proceedings of the IEEE Int.. Conf. on Comm.*, 2007, pp. 4646 – 4651.

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

[17] R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in UWB channels," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 364–375, 2007.

[18] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3776–3784, Nov 2005.

[19] A. Hassan, W. Stark, J. Hershey, and S. Chennakeshu, "Cryptographic key agreement for mobile radio," *Digital Signal Processing*, vol. 6, pp. 207–212, 1996.

[20] H. Koorapaty, A. Hassan, and S. Chennakeshu, "Secure information transmission for mobile radio," *IEEE Communication Letters*, vol. 4, no. 2, Feb 2000.

[21] T. S. Rappaport, *Wireless Communications: Principles and Practice.* Prentice Hall PTR., 2001.

[22] J. K. Tugnait, L. Tong, and Z. Ding, "Single-user channel estimation and equalization," *IEEE Signal Processing Magazine*, vol. 17, pp. 16–28, 2000.

[23] W. C. J. Jr., *Microwave Mobile "Communiations.* Wiley, 1974.

[24] T. Moore, "IEEE 802.11-01/610r02: 802.1x and 802.11 key interactions," *Microsoft Research*, 2001.

[25] S. Fortune, D. M. Gay, B. Kernighan, O. Landron, R. A. Valenzuela, and M. Wright, "Wise design of indoor wireless systems: practical computation andoptimization," *Computational Science and Engineering, IEEE*, vol. 2, no. 1, pp. 58–68, April 1995.

[26] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography.* CRC Press, 1996.

[27] U. M. Maurer, "A universal statistical test for random bit generators," *Journal of Cryptology*, vol. 5, pp. 89–105, 1992.

[28] NIST, "A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications," 2001.

[29] "IEEE standard 802.11a: Part 11 wireless LAN medium access control (MAC) and physical layer (PHY) specifications: High-speed physical layer in the 5 GHz band."

[30] Q. Wang, S. R. Kulkarni, and S. Verdu, "A nearest-neighbor approach to estimating divergence between continuous random vectors," in *Int. Symp. on Inform. Theory*, 2006, pp. 242–246.