Robust Key Generation from Signal Envelopes in Wireless Networks

Babak Azimi-Sadjadi*
Intelligent Automation, Inc.
15400 Calhoun Drive, Suite 400 Rockville, MD, USA
babak@i-a-i.com

Aggelos Kiayias
Department of Computer Science Engineering
University of Connecticut, Storrs, CT, USA
aggelos@cse.uconn.edu

Alejandra Mercado
Department of Electrical Computer & Systems Engineering, Rensselaer Polytechnic Institute, Troy NY, and
ADG, Hughes Network Systems, Germantown MD, USA
mercado@ecse.rpi.edu

Bulent Yener
Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA
yener@cs.rpi.edu

ABSTRACT

The broadcast nature of a wireless link provides a natural eavesdropping and intervention capability to an adversary. Thus, securing a wireless link is essential to the security of a wireless network, and key generation algorithms are necessary for securing wireless links. However, traditional key agreement algorithms can be very costly in many settings, e.g. in wireless ad-hoc networks, since they consume scarce resources such as bandwidth and battery power.

Traditional key agreement algorithms are not suitable for wireless ad-hoc networks since they consume scarce resources such as bandwidth and battery power.

This paper presents a novel approach that couples the physical layer characteristics of wireless networks with key generation algorithms. It is based on the wireless communication phenomenon known as the *principle of reciprocity* which states that in the absence of interference both transmitter and receiver experience the *same signal envelope*. The key-observation here is that the signal envelope information can provide to the two transceivers two correlated random sources that provide sufficient amounts of entropy which can be used to extract a cryptographic key. In contrast, it is virtually impossible for a third party, which is not located at one of the transceiver's position, to obtain or predict the exact envelope; thus retrieve the key. Since in the presence of interference strict reciprocity property can not be maintained; our methodology is based on detecting *deep fades* to extract correlated bitstrings. In particular, we show how a pair of transceivers can reconcile such bitstrings and finally flatten their distribution to reach key agreement. In our constructions we use cryptographic tools related to randomness extraction and information reconciliation. We introduce "secure fuzzy information reconciliators" a tool that enables us to describe robust key generation systems in our setting. Finally we provide a computational study that presents a simulation of a wireless channel that demonstrates the feasibility of our approach and justifies the assumptions made in our analysis.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication.

General Terms: Algorithms, Security, Theory, Measurements.

KeyWords: Physical layer security, randomness extraction, signal envelopes, wireless ad-hoc networks.

1. INTRODUCTION & BACKGROUND

Secure communications in wireless ad-hoc networks requires efficient key generation and update (renewal) algorithms which are essential to ensure (1) message confidentiality, (2) message integrity, and (3) node authentication. However, several characteristics of wireless ad-hoc networks make commonly used solutions for efficient key generation and update algorithms in wired networks inapplicable. Firstly, the wireless communication medium is in general a broadcast environment — anyone with a tuned receiver within a radius that permits adequate signal to interference and noise ratio (SINR) can *eavesdrop*. Secondly, network nodes often operate with limited battery and computation power, and

*This work was done in part while the author was with Rensselaer Polytechnic Institute.

CCS'07, October 29–November 2, 2007, Alexandria, Virginia, USA.

memory. Thirdly, wireless nodes may be mobile and the state information about their neighborhood may change — possibly often. Fourthly, nodes may not be able to access to public key infrastructure (PKI) for securing their communications. It is desirable to design key generation and renewal algorithms for wireless ad-hoc networks that will minimize message exchange. Currently, there are no algorithms to achieve key generation and renewal without exchanging messages and investing great computational cost. Existing key-generation algorithms such as Diffie-Hellman [18] are costly in terms of computation and communication and are designed independently from the physical characteristics of the networks where they will be executed.

The main contribution of this work is to couple the physical channel characteristics with key generation algorithms to secure wireless ad-hoc networks. The novelty lies in the robustness of this technique to ambient interference and to errors in the channel estimation. Our techniques exploit the *reciprocity principle* of wireless communications, which states that two transmitters working with the same carrier frequency, in the absence of interference (we relax this later) will experience *the same (relative) signal strength from each other at the same time*[1].

In practice, the presence of interference cannot be neglected in a wireless network and the reciprocity principle does not strictly apply. Yet the techniques presented here do not require identical signal envelopes for both parties, but only *matching deep fades*, which are impervious to reasonable levels of interference, i.e. SINR. By reasonable levels of SINR, we mean SINR levels that allow the communication link to have acceptable bit error rate (BER). We note that the acceptable SINR depends on the specific modulation technique. For example, if the target symbol error rate (SER) is 10^{-5} then for PSK modulation we require the SINR to be about 24 dB for a typical Rayleigh channel (i.e., the received signal power is 24 dB stronger than the combined receiver noise and perceived ambient interference). This means that the deep fades that can be measured go as far as -24 dB deep (that is, when the receiver predominantly perceives noise plus interference, the desired signal having dropped below those two). The modulation technique QAM 64 (that provides higher rates at the expense of greater sensitivity to noise,) will require an SINR of about 33 dB for the same SER. Therefore, detecting a deep fade even in the presence of noise and interference is possible.

In a typical environment, reflective surfaces vary from moment to moment (i.e., received signals are time-variant): a truck may be passing by a window, a reflective surface may tilt removing or adding multi-path, or the network node itself may be in a moving vehicle. Hence the fading characteristics are, in practice, very difficult to predict, and are usually modeled as a stochastic process. However, whatever realization of that process occurs for a network receiver, the signal it sends back to its counterpart will experience the same realization of that fading at that instant. Note also

[1]This is because the typical *fading* phenomenon is created by the transmitted signal bouncing off of various reflectors on its way to the receiver and all these *multi-path signals* arrive with phase offsets. When the phase difference produces destructive interference, the receiver experiences a deep fade. When the phase difference is small, the receiver experiences a strong signal. But the electromagnetic paths going from the transmitter to the receiver are the same as if their roles were reversed.

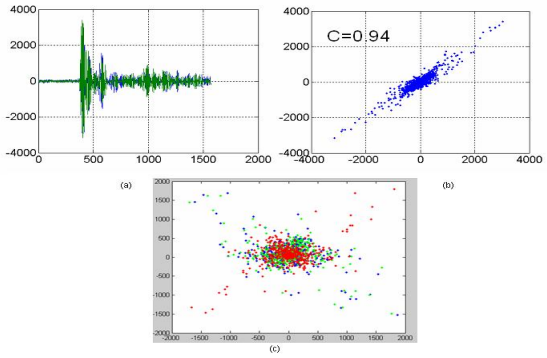

Figure 1: (a): Signal received by radio 1, and radio 2 vs. time. Vertical axis: proportional to voltage on antenna. Horizontal axis: time in units of 36ps. (b): Signal from radio 1 vs. signal from radio 2. The small deviations from a line through (0,0) with unit slope are caused by: 1) operator moving during data acquisition, and 2) Small differences between the radios. The signals are highly correlated, with a correlation coefficient of 0.94. (c): same as center (b), but with one of the radios moved to another room 20 feet away. We compare the second data set with one of previous sets. The multipath has changed dramatically, and only random correlations are left (C=0.1) Thus eavesdropping will be virtually impossible for an adversary unless it comes very close to the sender or receiver, but then it will be detectable.

that the phase differences of the arriving multi-paths are quite sensitive to the position. For example, for a carrier of 850 MHz, the wavelength is about a foot long, thus constructive interference (signal high) may change to destructive interference (deep fade) by shifting a mere half a foot. Thus, *a transceiver acting as an* ***eavesdropper****, in any other position will experience different fading characteristics.*

Figure 1 demonstrates the reciprocity measurement using two Ultra Wide Band (UWB) transceivers. As depicted in Figure 1, two transmitters experience the same (relative) signal strength, and the received signals at the receivers are highly correlated. Also it is clear from the figure that the eavesdropper's received signal has very little correlation with the received signal in the legitimate receivers. By passing the UWB signal through a filter with the bandwidth of the channel we get two signals (at both legitimate receivers). These two signals will have a deep fade at the same time instance.

The same phenomenon happens when the measurements are done in frequency domain (the frequency domain measurement is the dual of time domain measurement)as shown in Figure 2. The measurements are done at both legitimate receivers. As can be seen from the measured frequency response, the deep fades occur at the same frequency. The reason that some of the deep fades do not match in this figure is because the measurements are not done at exactly the same time (transceivers cannot transmit and receive simultaneously, but must allow for a small delay). So the change in environment appears in the measurement.

Our hypothesis is that these *fading graphs can be used to generate cryptographic keys, and the non-stationary characteristics of a wireless network can be used to extract enough*

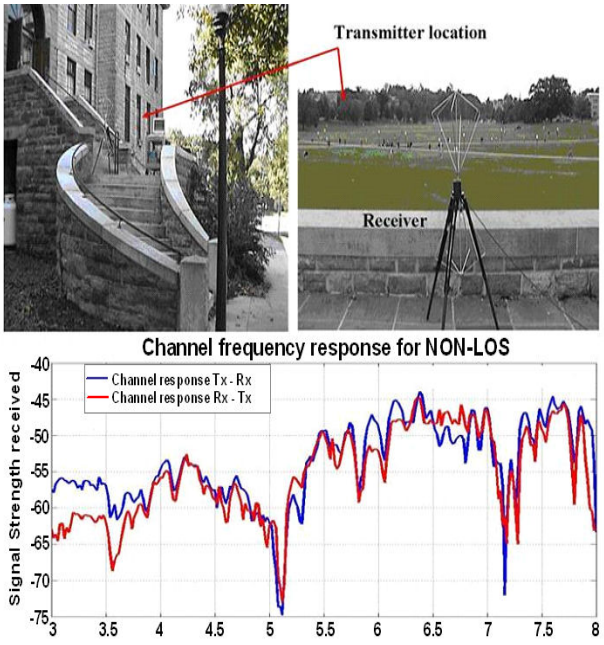

Figure 2: LOS measurement terrain. The channel power spectrum is reasonably flat with 2 null positions. The response is reciprocal.

entropy to obtain cryptographically secure keys. As long as the channel does not become permanently stationary, the keys can be renewed frequently to cope with playback attacks. Furthermore, once the first key is securely obtained, such properties can be used to ensure authenticity, and prevent man-in-the-middle and replay attacks.

To facilitate robust cryptographic key-generations two technical challenges must be met: (i) information reconciliation between the correlated random sources that are available to the two transceivers, and (ii) flattening of the key distribution for the purpose of extracting a high quality key. We introduce two methods for key generation based on: (1) error-correction and key-verification-information, and (2) the new primitive of fuzzy information reconciliators that we introduce.

We note that no special or added hardware beyond threshold detectors — which are already present in transceivers — is required, and the nodes use cheap and common omnidirectional antennae, and do not require smart antennae, or arrays.

1.1 Related Work

There is no one-size-fits-all key management scheme for all wireless networks (refer to [7] for a detailed survey). The proposed solutions depend on the network architecture, existence of trusted third parties, available resources on wireless clients and the capabilities of adversaries.

In ad-hoc wireless networks the general approach is to equip each node with either (i) a master key, or (ii) a list of keys (a key-chain), or (iii) keying materials; so that a pair of wireless nodes can either find a key in common, or generate it. In master key based solutions [29, 19], wireless nodes are pre-distributed a master key. Two nodes first exchange random nonce or node ID and use the master key along with a pseudo random function to generate a symmetric session key. In key-chain based solutions, each wireless node is pre-distributed a list of keys, called a key-chain. Two nodes just exchange their list of key IDs and use the combination of common keys as the symmetric session key. Key-chains must be carefully designed so that either two nodes have a key in common in their key-chains and they have a wireless link between them, or there is a path, called a key-path, among these two nodes where each pair of neighboring nodes on this path have a key in common.

Key pre-distribution approaches are outside the scope of this work, since they are not pertinent to our setting (that assumes no joint node setup). Algorithms to generate the key-chains fall into one of the three classes: (*i*) probabilistic [20] [12] where key-chains are randomly selected among a pool of keys, (*ii*) deterministic where key chains are designed from a set of keys by using algorithms such as Balanced Incomplete Block Design (BIBD) of design theory [8, 10, 9], and (*iii*) hybrid probabilistic and deterministic schemes [8]. In dynamic key generation solutions a set of public and private keying materials is formed in a probabilistic, deterministic or hybrid manner and is pre-distributed to each wireless node. Two nodes exchange their public information such as node ID in a polynomial based solution [4] or public column vector in matrix based solution [3].

The concept of combining key management and physical layer characteristics is first presented in [21]. More recently (independently from our research) in [2] which uses steerable parasitic array radiator antennae in contrast with our method that requires only ubiquitous and cheap omnidirectional antennae. Furthermore, their method relies on strict reciprocity, with no more distortion than noise and differences in transmission powers. In a real network, the most pernicious presence causing distortion is interference, not noise, which is often orders of magnitude lower than interference. This breaks down reciprocity, which is why our method addresses this problem by focusing on the deep fades, rather than the entire envelope. In [31] communication between an Access Point and a User Terminal is considered. This approach requires also steerable parasitic array radiator antennae. In addition to the special antenna, that technique requires overhead bandwidth expenditure in that the Access Point must transmit a constant amplitude wave, which serves no purpose other than generating the key. Once again, strict reciprocity is required for the uplink and downlink signal profiles to match. In a practical setting with interference present, that simply won't be available. Another method which is based on the time-varying frequency characteristics, and is suitable for OFDM systems is proposed in [28]. It utilizes channel reciprocity and the time-variant frequency characteristics to generate a security key. It also measures time difference compensation of the channel and uses a synchronous addition process for noise reduction to prevent errors in key generation, which is different from ours. In comparison, our approach is much more inexpensive and less sensitive to estimation errors. *Overall, compared to all these previous works, our approach is an improvement, as it eliminates message exchanges, special antennas, strict reciprocity assumptions and does not limit itself to UWB communication.*

Our approach takes advantage of cryptographic tools that relate to randomness extractors, [32, 34] and fuzzy extractors [14]. Key agreement in our work targets the setting

where the two parties that wish to exchange a key have access to two correlated random sources (the deep fade information derived from the channel envelope) while the adversary has only partial access to this source. Key agreement with restricted adversaries has been studied theoretically in the works of Maurer, [26], Maurer Wolf, [27] [22] under minimum entropy assumptions and specific restrictions imposed on the correlation of the two sources (e.g. agreement with high probability); non formal approaches appeared much earlier [35] and [13]. Our work differs from these previous works since we are using the specifics of our setting and we thus we need to error-correct as well as flatten the key distribution taking into account the specifics of our channel characteristics. A different line of works have studied other type of resource-bounded adversaries in terms of memory is [1, 6] (whereas the adversarial restriction in our case is knowledge of correlated random source). Our primitive of secure fuzzy information reconciliators is related (and inspired) from the work of [14, 17]; it differs from the notion of the fuzzy extractor as it is only requires to work for a specific error type and metric (and thus it needs not the generality of a fuzzy extractor). Indeed the metrics of similarity considered in [17] are not suitable for our methods as those are motivated from biometric key generation (cf. [24, 25]) and do not apply to our domain. The appropriate metrics for fuzzy extraction in our domain resemble error vectors that are encountered in the setting of shift-error correction systems, cf. [23], and thus our "information reconciliation" strategy is suitable for such error patterns and their corresponding metric.

2. SAMPLING THE RANDOM SOURCE

In our approach secret keys are generated periodically by detecting deep fades in the data transmission between both transceivers. Each transceiver samples its random source the signal it receives and checks to see if each sample exceeds a agreed-upon threshold for *deep fades*. Although signal envelopes of fading channels may change due to interference, the probability of detecting a false positive or missing a deep fade is low for practical systems with reasonable average SINR levels (we argue about this with simulation results in section 4). Thus, we can utilize deep fades in the received signal envelopes in Time Division Duplex (TDD) systems — which distinguish uplink and downlink messages by using different time slots — to extract some correlated random variables at the two transceivers.

2.1 Background: Fading channels and reciprocity

We present a brief overview on the reciprocity and channel fading. Fading is caused by multi-path propagation and its variation is caused by the mobility of one or both the transmitting and receiving nodes (or their environment). The randomness in the signal's envelope is caused by path loss (due to the distance between the nodes), knife-edge diffraction (often caused by the corner of a building), shadowing loss (by obstructions), and fading loss (due to the multipath described above). Usually some (or many) of these causes are time-varying; more so when the network nodes themselves are mobile.

The moving speed of the nodes causes a doppler frequency shift and the signal power spectrum spreads over the frequency domain. With a moving speed of V and a signal wavelength λ, the doppler frequency is $f_d = \frac{V}{\lambda}$.

The doppler shift results in what is called a time-selective channel. This time-selective property is approximated by the *coherence time*, $T_c = \frac{9}{16\pi f_d}$.

The coherence time is the time duration over which a received signal's amplitude and phase are predictable. The channel impulse response is invariant during the coherence time. If the symbol time of the received signal is smaller than coherence time, then the channel is called time-invariant. The coherence time is used to define the channel fading characteristics in the time domain.

2.2 Thresholding

The two transceivers will use the channel fading information to extract a bit stream (that will later be used for key generation). The bit stream is generated based on a threshold that is set by both sides of the wireless link. The statistics of the generated bit stream and consequently the generated key depends on this threshold as well as the transmit power and the attenuation in the link. To determine this threshold an automatic gain control (AGC) mechanism can be used so that the statistics of the generated key is independent of the transmit power and the link attenuation.

The occurrence of a fade and its duration is a random process. Once the threshold is set, the average fade duration and level crossing rates depend on the channel statistics [5]. For a Rayleigh fading channel it is shown that the mean fade duration and the level crossing rates are given as follows:

$$\bar{\tau}(R) = \frac{e^{\rho^2} - 1}{\rho f_m \sqrt{2\pi}}$$

where $\rho = \frac{R}{R_{rms}}$ and f_m is the maximum Doppler frequency, R is the threshold, and R_{rms} is the RMS value of the received signal. The rate of occurrence of fades (signal crossing threshold R) is given by

$$N(R) = \sqrt{2\pi} f_m \rho e^{-\rho^2}$$

Consider the scenario where node A transmits its signal to node B while receiver C (an adversary) is listening to the same broadcast. If C is more than a wavelength away from B, then the occurrences of deep fades at B and C are independent. Therefore, the adversary cannot guess the exact moment of deep fade occurrences or their duration.

2.3 Deep Fades to Bit Vectors

The next step after selecting a fade crossing threshold for the signal envelope is to compare the received signal envelope over each time slot with said threshold. If the envelope of the received signal is below the threshold, which means a deep fade occurred, we set a bit to 1 for this time slot. Conversely, if the envelope of received signal is above the threshold, which means no deep fade happened over this time slot, we set a bit to 0 for this time slot. After a period of time, a bit stream from each downlink and uplink channel is obtained to construct the bit vectors (BV). The bit vectors from the downlink and from the uplink channels are quite similar because they receive signals with similar characteristics due to channel reciprocity. *Although the downlink node and uplink node access the channel in different time slots, channel reciprocity results in similar channel response for both as long as the duration of each time slot is much smaller than the channel coherence time.*

One important innovation of our system is that the key

generation circuit passes the received signal through a very-narrow-band filter for a narrow-band system, or through a bank of several very-narrow-band filters if the channel is frequency selective. In the former case, many narrow-band interferers are likely to be filtered out entirely. This is a very economical way for both cases to reduce the effect of interference (or even an adversary's jamming signal).

2.4 The Random Source Characteristics

Given the above it follows that the two transceivers will be capable of retrieving two bitstrings that will have a number of "runs" (sequences of 1's) corresponding to the deep fades they experienced in their signal envelope.

The bitstrings would be correlated due to the reciprocity principle but they will also have a number of discrepancies. For example, there will be a discrepancy at the beginning or the end of each deep fade if the deep fade lasts over a number of time slots. Another reason for bit discrepancy is because the stream in the downlink may be a slightly shifted version of the one in the uplink. Yet another reason for discrepancy is to have one of the two transceivers believing that aÊcertain deep fade occurred over some time slots where the other transceiver has no such information (such discrepancy is due to chattering and/or other local noise conditions). We will deal with such discrepancies in two different ways: we will apply error-correction (or information reconciliation techniques) to correct shift type of errors; chattering on the other hand, will be dealt with filtering. The adversary in all cases is assumed to have the information on the number of deep fades that have occurred in a certain time-frame but he will not be privy to the locations of such fades.

3. KEY GENERATION

Let A and B be the two parties that wish to generate a key; we abstract the problem as follows. The two parties have access to two correlated random sources R_A and R_B over $\{0,1\}^n$; in addition to the two parties, we also assume the existence of an adversary that may eavesdrop or even interfere with the random sources R_A and R_B. Whenever A and B sample their random sources, R_A and R_B, they obtain two bitstrings ρ_A and ρ_B respectively. Moreover, the adversary obtains a bitstring ρ_C. The triple of random variables (ρ_A, ρ_B, ρ_C) is distributed according to Env, a joint distribution that is based on the properties of the channel as well as assumptions about the environment that affect the wireless transmission. In some settings the adversary will have no information whatsoever about ρ_A, ρ_B; this translates to the setting where the variable ρ_C is independent of the variables ρ_A, ρ_B. In our approach we use tools such as randomness extractors and the leftover hash lemma – we refer to [17] for a comprehensive survey. We briefly recall below the notion of a (randomness) extractor:

DEFINITION 3.1. Randomness Extractor*: a function* Ext *is called a* (n, m, l_0, ϵ)*-extractor if* Ext *is a mapping* $\{0,1\}^n \times \mathcal{R} \rightarrow \{0,1\}^{l_0}$ *such that if* ρ *is* any *random variable satisfying* $H_\infty(\rho) \geq m$ *it holds that* $||\langle \mathsf{Ext}(\rho, \tau), \tau\rangle - \langle \rho_{\mathsf{u}}, \tau\rangle|| \leq \epsilon$*, where* ρ_{u} *is uniformly distributed over* $\{0,1\}^{l_0}$ *and* τ *is uniformly distributed over* $\mathcal{R}$*. Alternatively, if* ρ *is a specific random variable and the function* Ext *satisfies the above property, we will say that* Ext *is a* $\langle n, l_0, \epsilon\rangle$*-extractor for* ρ*.*

We next formally define our notion of a key exchange system in our setting . We require three properties: (i) correctness, which ensures that both parties end up with the same key with high probability, (ii) uniformity, which ensures that keys' distribution are close to uniform, and (iii) security, which ensures that no adversary can compute with substantial probability an arbitrary chosen function of the key given the transcript information.

Formally, a $(n, l_0, \epsilon_{\mathsf{c}}, \epsilon_{\mathsf{u}}, \epsilon_{\mathsf{s}})$-key-generation-system is a pair $(\mathsf{KG}, \mathsf{Env})$ where (1) Env is a product probability distribution $\langle \rho_A, \rho_B, \rho_C\rangle$ over $\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$, (2) KG is a two-party protocol that returns private output in $\{0,1\}^{l_0}$ for both players (that are polynomial-time bounded in n) and (3) the following three properties are satisfied (note that PPT stands for probabilistic polynomial time):

DEFINITION 3.2. **Correctness.** *If* (ρ_A, ρ_B, ρ_C) *is a random variable distributed according to* Env*, it holds that the event that both players return the same output in the protocol* KG *is at least* $1 - \epsilon_{\mathsf{c}}$*. Note that correctness does not take into account the random variable* ρ_C*.*

Uniformity. *If* (ρ_A, ρ_B, ρ_C) *is distributed according to* Env*, it holds that the statistical distance of the output key of player A from the uniform distribution over* $\{0,1\}^{l_0}$ *is at most* ϵ_{u}*.*

Security. *Given any PPT* $\mathcal{A}$ *there is a PPT* $\mathcal{A}'$ *that satisfies the following for any function* f*:*

$$|\mathbf{Prob}[\mathcal{A}(\mathsf{t}) = f(key_{\mathsf{t}})] - \mathbf{Prob}[\mathcal{A}'(1^n) = f(key_{\mathsf{t}})]| \leq \epsilon_{\mathsf{s}}$$

where t *is distributed over the* KG *transcripts and* key_{t} *is the key of player A that corresponds to transcript* t*.*

We note that the definition above is in a passive sense (the adversary is eavesdropping honest interactions); this can be generalized to active security but we defer such extension for an upcoming work. The definition of security, parallels "entropic-security" as defined in [15]. Also note that the requirements of uniformity and security can be modeled together but for clarity they are separated in the formalization above.

3.1 Key Generation based on Key Verification Information

In this section we will present a first method for designing a key-generation system. Recall our main observation that the differences between the random sources, R_A and R_B, that are observed between the two legitimate players will predominantly happen at the beginning and(or) at the end of some deep fades, and that such fades occur randomly over a period of time. In this section we exploit the fact that, based on some key-verification information released by one of the two players, the other player may correct such differences. Without loss of generality, let us differentiate the two players A and B calling A the sender and B the receiver. The approach is as follows: suppose that $\langle \rho_A, \rho_B, \rho_C\rangle$ is distributed according to Env; now assume that ρ_A is the "correct" bitstring[2], i.e., the sender has the correct bitstring, and the receiver has to correct its own bitstring ρ_B to match ρ_A.

[2]We use quotation marks because, in reality, neither player may have a bit vector reflecting the *true* physical channel realization. Yet this does not concern us, since we only care that both players have matching keys, not that said key should flawlessly reflect the channel.

Given any bitstring, a *run* is a sequence of consecutive of 1's within the bitstring. Based on the mapping of deep fades into sequences of 1's, it is clear that ρ_A contains a run for each deep fade that occurred in the envelope of the wireless transmission. Suppose the length of each ρ_A and ρ_B bitstring is n, the number of deep fades is t, and each fade extended for a number of k_τ time slots ($\tau = 1, \cdots, t$), i.e., each fade resulted in a k_τ-bit long run within the string ρ_A. To simplify the analysis of this section, we assume that in n time-intervals there exist t deep fades, each one of length k; note that in reality the parameter k varies for each fade (cf. Section 4) but this will not affect substantially the analysis we present here (and in fact we drop this assumption in the system of section 3.2)

Given our assumption that deep fades are uniformly distributed within the time interval (and assuming for now that t and k are fixed constants, and $n \geq 2kt$), the entropy $E_{n,t,k}$ of the string ρ_A will be at least:

$$E_{n,t,k} \geq \log_2 \prod_{l=1}^{t} (n-k+1-(l-1)(2k-1)) - \log_2(t!)$$

where the product above denotes the number of ways to arrange t runs of length k within a bitstring of length k where the subtraction of the $t!$ is due to the fact that the order of the placement of such runs is of no importance. Now observe that for the multifactorial function $m!^{(v)} = m(m-v)(m-2v)\ldots$ it holds that: $v^{\lfloor m/v \rfloor}(\lfloor m/v \rfloor)! \leq m!^{(v)} \leq v^{\lfloor m/v \rfloor + 1}(\lfloor m/v \rfloor + 1)!$. Based on this, we obtain

$$E_{n,t,k} \geq \log_2(v^{\lfloor m/v \rfloor} \cdot (\lfloor m/v \rfloor)!/(v^{\lfloor m/v \rfloor - t}(\lfloor m/v \rfloor - t)!)/t!)$$

for $m = n-k+1, v = 2k-1$. Next, from Stirling's approximation we have $\sqrt{2\pi}n^{n+1/2}e^{-n+1/(12n+1)} < n! < \sqrt{2\pi}n^{n+1/2}e^{-n+1/(12n)}$ from which we can obtain the bound for the falling factorial

$$(q)_w = q!/(q-w)! > \frac{1}{e}(\frac{q}{q-w})^{q+1/2}(\frac{q-w}{e})^w$$

. Using this we obtain:

$$E_{n,t,k} > \log_2\left(\frac{1}{e}\frac{(2k-1)^t}{t!} \cdot \alpha \cdot \beta\right)$$

where

$$\alpha = \left(\frac{\lfloor \frac{n-k+1}{2k-1} \rfloor}{\lfloor \frac{n-k+1}{2k-1} \rfloor - t}\right)^{\lfloor (n-k+1)/(2k-1) \rfloor + 1/2}$$

and

$$\beta = \left(\frac{\lfloor \frac{n-k+1}{2k-1} \rfloor - t}{e}\right)^t .$$

Based on the above the following theorem is proved:

THEOREM 3.3. *It holds that*

$$E_{n,t,k} = \Omega(t \log k + (n/k)\log(n/(n-kt)) + t\log(n/k-t))$$

Notice that t and n/k are the dominant asymptotic terms that control the amount of entropy of $E_{n,t,k}$.

In order to achieve agreement between the two parties, we take advantage of the fact that the runs of ρ_A and ρ_B may be different only in the beginning and ending bits of a deep fade. Suppose that s is a parameter that specifies

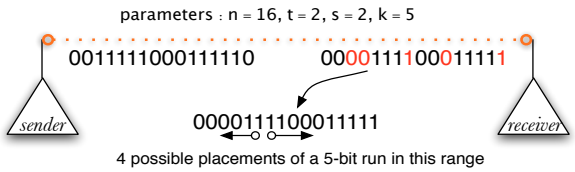

Figure 3: A representation of direct key generation by searching all possible keys.

the maximum number of bits that can be different on either side of a run between ρ_A and ρ_B. Recall that we assume that the sender has t runs of length k. Now suppose that for some parameter s it holds that $k > 2s$, and when ρ_B is sampled each run may be extended to the left or right by a number between zero and s bits. This suggests that if a run is observed in the interval $[f, g]$ by the sender it holds that there exist $r, l \in \{-s, \ldots, s\}$ such that the receiver observes the same run at locations $[f+l, g+r]$ and the length of this run is also k, i.e., $r - l = k + f - g - 1$. The total number of pairs (r, l) that satisfy the constraint is $2s+1$, so for each run the receiver has a number of $2s+1$ possibilities. Given that there are t runs we have that, the total space of errors includes at most $(2s+1)^t$ vectors; see figure 3. For reasonably small values of s, t this means that it is possible for the receiver to scan through all possibilities and recover the exact bitstring that was obtained by the sender. Note that keeping t small will not necessarily make the entropy of the channel too low as we can still rely on the value of n/k to maintain it at a safely high level for cryptographic key generation cf. theorem 3.3. We remark that the analysis when there is small variation in k from one run to the other follows a similar approach. From this discussion, it follows that the receiver will require some "key verification information" so that it is assisted in finding the correct match.

Given the above, the key-generation algorithm will operate as follows:

(1) the sender and receiver will sample ρ_A and ρ_B, respectively;

(2) the sender then, will calculate the key and send a *key verification information* (KVI) to the receiver;

(3) based on KVI the receiver decides on the correct key by scanning through all possible error-vectors.

The key verification information submitted to the receiver is obtained by computing a value of the form $\langle \mu, \kappa, \mathcal{U}_\kappa(key) \rangle$ where μ, κ are selected at random from a fixed bitstring size and $\mathcal{U}$ is a keyed hash function to be specified below in theorem 3.4; finally, *key* is computed as $key = \mathcal{H}(\mu, \rho_A)$ where $\mathcal{H}$ is a hash function to be specified below in theorem 3.4. The receiver, using ρ_B, tries all $(2s+1)^t$ modifications and attempts to match the $\mathcal{U}_\kappa$ value using $h_{\mathsf{i}} = \mathcal{H}(\mu, \rho_B^{\mathsf{i}})$ as the key where $\mathsf{i} = 1, \ldots, (2s+1)^t$ and ρ_B^{i} is the i-th possible version of ρ_B. If there is a match, the search stops and the receiver sets his key as h_{i}.

Suppose now that the adversary, through knowledge of the statistics of the channel, deduces the average number of deep fades t as well as their average length, k — he cannot know their locations. We show the following:

THEOREM 3.4. *Assume the following three conditions:*

1. *Suppose* $key \neq key' \in \{0,1\}^{l_0}$*, then it holds*

   $\mathbf{Prob}[\mathcal{U}_\kappa(key) = \mathcal{U}_\kappa(key')] \leq \epsilon_2$ *where* κ *is distributed uniformly over* $\{0,1\}^k$*. In other words,* $\{\mathcal{U}_\kappa\}_\kappa$ *is a* **universal hash family**.

2. $\mathcal{H} : \mathcal{R} \times \{0,1\}^n \rightarrow \{0,1\}^{l_0}$ *satisfies that the random variable* $(\mu, \mathcal{H}(\mu, w))$ *has* ϵ_1 *statistical distance from* (μ, u) *that is uniformly distributed over* $\mathcal{R} \times \{0,1\}^{l_0}$ *and* w *distributed according to* R_A *conditioned on* R_C*. In other words,* $\mathcal{H}$ *is an* $\langle n, l_0, \epsilon_2 \rangle$**-extractor** *for the source* R_A *conditioned on* R_C*.*

3. *The probabilistic map* $\mathsf{F}(w) = (\kappa, \mathcal{U}_\kappa(w))$ *with* κ *uniformly distributed,* **hides all functions** *of its input, i.e., for every PPT* $\mathcal{A}$ *there is a PPT* $\mathcal{A}'$ *such that for any* f*,* $\mathbf{Prob}[\mathcal{A}(\mathsf{F}(w)) = f(w)] - \mathbf{Prob}[\mathcal{A}'(1^n) = f(w)]| \leq \epsilon_3$ *where* w *is uniformly distributed over* $\{0,1\}^n$*.*

Given the above, it holds that KG *described above is a* $(n, l_0, \epsilon_1, \epsilon_2, \epsilon_2 + \epsilon_3)$*-key-generation-system.*

The above theorem suggests that we can construct a key-generation system as long as the functions $\mathcal{H}$ and $\mathcal{U}$ satisfy the stated properties. First, $\mathcal{U}$ has to be a universal hash function family [11, 34]. Second $\mathcal{H}$ must be an extractor for the source R_A, i.e., given the random variable ρ_A that is distributed according to the triple $\langle \rho_A, \rho_B, \rho_C \rangle$ from Env, it holds that $\mathcal{H}(\mu, \rho_A)$ is ϵ_2 away from the uniform distribution of $\{0,1\}^{l_0}$. This needs to happen conditioned on R_C as prescribed in the distribution of Env. To implement this function we can use a general purpose extractor that can be constructed based on universal hash functions; this a standard construction that also applies to the case of conditional entropy as shown in [17]. In this case it will hold that $l_0 = E_{n,k,t} + 2 - 2\log\epsilon_2^{-1}$ where $E_{n,k,t}$ is the entropy function defined in theorem 3.3. Finally regarding security, the function $\mathcal{U}$ needs to additionally (to being a universal hash) to also hide all functions of its input, (this is possible as e.g., described in [15]). We note that it would also be possible to "throw away" the bits of key that are fixed by $\mathcal{U}_\kappa(key)$ and use the remaining bits; this would improve security but it would reduce the efficiency of the scheme (as we would need to extract more bits).

To illustrate the feasibility of the approach, we provide the details of an (ad-hoc) implementation of the construction:

Example Implementation. As seen from the arguments leading to Theorem 3.3, we have that for $k = 5, t = 12, n = 512$ it holds that the conditional entropy of ρ_A given ρ_C is at least 77 bits. Using a universal hash family for $\mathcal{H}$ we can obtain a 55-bit key that is 2^{-12} away from the uniform distribution over $\{0,1\}^{55}$. In order for the receiver to recover this key for $s = 2$, it will have to execute a brute-force step of 2^{24} operations, where each one involves one application of the universal hash family $\mathcal{H}(\nu, \cdot)$ and one application of the universal-one-way hash $\mathcal{U}_\kappa(\cdot)$. If $\mathcal{H}$ is substituted with a universal hash of comparable time complexity to that of MD5 and $\mathcal{U}$ is substituted with a universal one-way hash family comparable to an HMAC, we have that the key can be recovered in at most 42 seconds in a standard laptop[3].

[3]Based on `openssl` benchmarks on a Macbook Pro that performs: (i) 2043780 HMAC(MD5) operations in 2.98 seconds, (ii) 2673300 MD5 operations in 2.98 seconds.

3.2 Key generation using Secure Fuzzy Information Reconciliators

The solution of the previous section has the major shortcoming that the brute-force error-correcting step requires too much time to be completed, thus making the protocol inefficient. Moreover, in the analysis we assumed that the length of each deep fade is the same. In this section we remove these two restrictions by presenting a key-generation system that relies on secure fuzzy information reconciliators (SFIR), a primitive we introduce here. We will show how SFIR can be instantiated and using such primitive we will present a key generation system that will enable very fast error-correction that is unconditionally secure; moreover, our approach in this section will work independently of the lengths of the deep fades. The benefits of the approach will come at the expense of sacrificing some additional bits of entropy.

We recall that a fuzzy extractor [14] is similar to a randomness extractor but it has a built-in error-correcting capability: any value of an imperfect random source that belongs to a sphere of a certain fixed radius for a given metric can be repaired to the same identical randomness extraction (given some helping information).

Below we define a variation of the fuzzy extractor primitive that is more suitable to our setting (to be explained below). We call our primitive a "secure fuzzy information reconciliator" or SFIR.

DEFINITION 3.5. *Let* $\mathsf{Env} = \langle \rho_A, \rho_B, \rho_C \rangle$ *be a joint random variable over* $\{0,1\}^{3n}$*. A* $(n, l_0, \epsilon_1, \epsilon_2)$*-secure-fuzzy-information-reconciliator (SFIR) for* Env *is a pair* $(\mathsf{Gen}, \mathsf{Rep})$ *that satisfies the following: (1) if* $\langle f, p \rangle \leftarrow \mathsf{Gen}(\rho_A)$*, then it holds that* $\mathbf{Prob}[\mathsf{Rep}(\rho_B, p) = f] \geq 1 - \epsilon_1$*. (2) the first output* f *of* Gen *is* ϵ_2 *away from the uniform distribution over* $\{0,1\}^{l_0}$ *conditioned on* ρ_C *as well as the second output* p *of* Gen*.*

The goal of this section is to design a SFIR scheme and then employ it to design a key agreement system, that will enable the sender and the receiver to recover the same key, $key = f$, even if they have slight discrepancies in their bit-vectors due to interference. Note that the definition of a fuzzy-extractor as given in [14] would not be a good match for our setting as we have a-priori knowledge about the error-distribution and it is unnecessary to mandate the min-entropy requirement as it is the case for a fuzzy extractor. Moreover, the metrics considered in [14, 17] are not suitable for our setting: the type of errors considered there, such as those that correspond to the Hamming or edit distance are more suitable for general error-correction of biometric key extraction, [14]. On the other hand, here, we need to correct a different class of errors that correspond to the shifts present in the runs within one of the two bitstrings (relative to the other). Finally, note that we need to incorporate a type of security into the definition of SFIR (hence the "secure" designation) : we require that the reconciliation information p still leaves sufficient entropy in ρ_A to extract a random key despite that the adersary knows additionally the correlated information ρ_C; we note that it is possible to define security in a more general way but this definition will be sufficient for our purposes now.

Our construction. The interpretation of the random variable that is produced by the envelope that will be used in this section is as follows: given the random pattern ρ, one of the

two parties (generically called the sender), records the values $\{\ell_1, \ldots, \ell_t\} \subseteq \{1, \ldots, L\}$ which are the locations of the deep fades within the L time slots. Note that $\ell_i \in \{0,1\}^u$ with $u = \lceil \log L \rceil$.

Our SFIR $\langle \mathsf{Gen}, \mathsf{Rep} \rangle$ uses an error-correction parameter s and operates as follows. Gen given ρ_A, computes the values $\mathsf{loc} = \{\ell_1, \ldots, \ell_t\}$ and then calculates the tuple $\langle \tilde{\ell}_1, \ldots, \tilde{\ell}_t \rangle$ where $\tilde{\ell}_j = \ell_j \bmod (2s+1)$. Then, Gen simply selects μ to seed an extractor $\mathcal{H}$ and produces the output $(f, p) = (\mathcal{H}(\mu, \overline{\rho_A}), (\mu, \langle \tilde{\ell}_1, \ldots, \tilde{\ell}_t \rangle))$; note that $\overline{\rho_A}$ is based on ρ_A but it is normalized so that all its runs are of length k where k is some fixed parameter (and thus note that $\overline{\rho_A}$ is not necessarily of length n); as we will see later this will not prohibit the reconstruction of $\overline{\rho_A}$ by the other transceiver.

The function Rep operates as follows: it receives as input ρ_B as well as the value $p = (\mu, \langle \tilde{\ell}_1, \ldots, \tilde{\ell}_t \rangle)$. The receiver will parse ρ_B for the locations of the deep fades and will find their locations $\{\ell'_1, \ldots, \ell'_t\} \subseteq \{1, \ldots, n\}$. It wil then attempt to correct to the original locations $\ell_1, \ldots, \ell_t$ by computing

$$\ell^*_j = \ell'_j - (\ell'_j \bmod (2s+1)) + \tilde{\ell}_j$$

Subsequently, Rep calculates a bitstring ρ^* with t runs of length k at locations $\ell^*_1, \ldots, \ell^*_t$. Then, Rep will feed μ, ρ^* into the extractor $\mathcal{H}$ and will terminate returning $f^* = \mathcal{H}(\mu, \rho^*)$. Observe that as long as $|\ell_j - \ell'_j| \leq s$ then it holds that $\ell^*_j = \ell_j$ and thus $\rho^* = \overline{\rho_A}$ and thus key agreement is achieved.

LEMMA 3.6. *The average min-entropy $\tilde{H}(\rho_A \mid \rho_C, p)$ where p is defined from $\langle f, p \rangle \leftarrow \mathsf{Gen}(\rho_A)$ is at least $D_{n,t,s} = \log \binom{n}{t} - t\lceil \log(2s+1) \rceil$.*

We remark that it is also possible to drop the least significant bit information from the fade locations to achieve agreement on a joint bitstring (so in this case the coordination information p would only need to agree on how many bits to remove). In the proof of the following lemma we rely on the leftover hash lemma [33] to implement the extractor $\mathcal{H}$:

LEMMA 3.7. *Suppose that $\langle \rho_A, \rho_B, \rho_C \rangle$ is distributed according to Env. Assume that with probability $1 - \epsilon_1$, ρ_A contains t runs and ρ_B contains t runs that shifted either left or right by an amount of s time slots. Then there is a way to implement $\langle \mathsf{Gen}, \mathsf{Rep} \rangle$ as above so that it is a $(n, l_0, \epsilon_1, \epsilon_2)$-SFIR with $l_0 = D_{n,t,s} + 2 - 2\log(1/\epsilon_2)$.*

Based on any SFIR $(\mathsf{Gen}, \mathsf{Rep})$, we define the following key-generation system:

Key Generation System based on a SFIR $(\mathsf{Gen}, \mathsf{Rep})$.

(1) The sender A will apply Gen to the random variable ρ_A to obtain a pair of strings $\langle f, p \rangle$; it will set $key = f$.

(2) the sender A will transmit to the receiver B the value p.

(3) the receiver B employs the function Rep and his reading of the envelope ρ_B to recover $key = f$.

Based on the SFIR properties we can easily show the following theorem:

THEOREM 3.8. *Given a $(n, l_0, \epsilon_1, \epsilon_2)$-SFIR for the envelope distribution $\langle \rho_A, \rho_B, \rho_C \rangle$, the protocol KG described above is a $(n, l_0, \epsilon_1, \epsilon_2, 0)$-key-generation system for the distribution $\mathsf{Env} = \langle \rho_A, \rho_B, \rho_C \rangle$.*

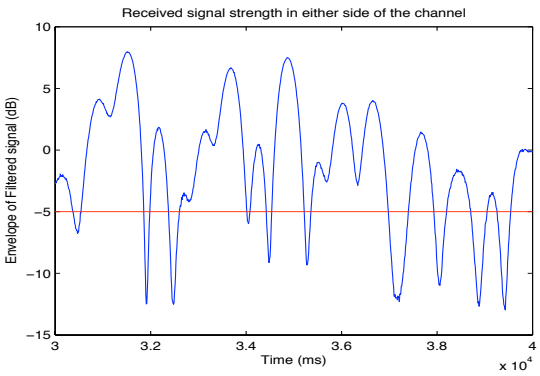

Figure 4: This figure shows one second the received signal strength at both sides of the communication channel after applying a low pass filter. The low pass filter is used to reduce the noise.

Example Implementation. Using the lemma 3.6, we have that for $t = 38, n = 2000, s \leq 4$ it holds that the min entropy is $D_{2000,38,4} = 191$. Based on the leftover hash lemma [33] we can obtain a key that is of length $l_0 = 81$ bits that will have distance less than 2^{-56} from the uniform distribution over $\{0,1\}^{l_0}$. Note that instantiating the leftover-hash-lemma is very simple (e.g., using linear mappings over finite fields, cf. [34]); it follows that the computational cost of the key-generation of this section is minimal.

Finally note that the parameters used in this example are consistent with the simulation results that we present in the next section (that dictate a fade rate of 19/1000 and a bound of s that is less than 4).

4. SIMULATION RESULTS

In this section we provide a simulated realization of two nodes transmitting signals through a Rayleigh fading channel, each receiving their own version of the signal, and extracting a bit vector from it. We then compare the two vectors to each other and show that with overwhelming probability the errors introduced in the communication links will be correctable based on our procedure that we described in 3.2.

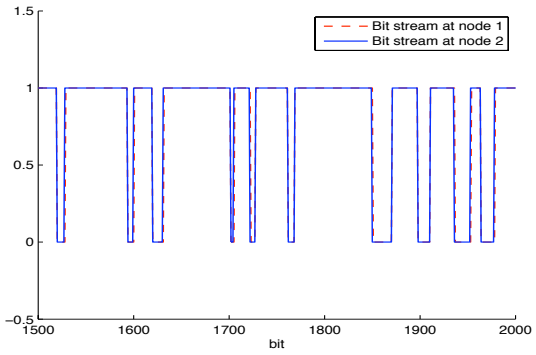

Figure 5: Comparison of the generated bits at node 1 and node 2.

4.1 Wireless Channel Simulation

We simulate a communication system with a Rayleigh fading channel, that both legitimate nodes experience. From each one's perception of a signal transmitted through this channel, they will generate their own bits streams. The parameters of this channel are:

(1) BPSK communication with the bit rate of 1 Mbps.

(2) SINR of 25 dB (equivalent to a BER of 10^{-5} for a multi-path fading channel).

(3) Doppler shift of 1 Hz.

(4) To reduce the effect of noise in estimating the bit streams at both sides of the channel, we filtered the received signal with a narrow low pass filter with a bandwidth of 100 Hz. Using a very narrow filter has the benefit of reducing the noise dramatically. Figure 4 shows the received signal strength of both sides of the communication channel after the low pass filter has been applied.

Note that in key generation we are only interested in estimating the received signal strength and not the actual transmitted bits. Also note that the Doppler frequency is in the order of a few Hz (at most 20 Hz for very fast changing environment), therefore, a narrow band filter with bandwidth of 100 Hz is enough to capture the signal fluctuation due to the change in the environment.

4.2 Generating bit streams

To generate bit streams in each side of the channel, each node samples the output of its low pass filter and compares it with a set threshold. Figure 5 shows the generated bits at both nodes when the threshold is set to -5 dB. As can be seen from the figure, the two generated sequences are very similar, in spite of the fact that each node experienced its own levels of interference and neither communicated with each other any decision regarding the generation of these bit streams. The only occasional differences occur when there is a transition from 0 to 1 or 1 to 0 — that is, at the edge of a deep fade. These mismatches between the sequences are due to many reasons, including the different timing between the two nodes (since there is a slot delay between each one's transmission) and each node's distinct interference and noise that passes through its low pass filter.

Note that Figure 5 depicts the *raw output of the low pass filter and threshold detector,* without engaging in any aforementioned techniques to match the two bit vectors up.

For the setup in this simulation, from a study of 100 seconds, deep fades occur with an average rate of 19 per one thousand bits[4]. This means that for $n = 1000$, the resulting number of fades is $t = 19$. Note that this simulation, using an actual Rayleigh fading channel, shows that k, the length of the run of 1s due to a deep fade, is a random variable, as was detailed in Section 2.2.

Hence these results confirm that *even in the presence of interference* in a wireless network with time division duplex (TDD) for communication, the similarity between envelopes of the transmitter and receiver is enough to obtain equal keys for both.

[4]The statistical data is extracted from long runs of the simulation explained in this section. The figures only show a portion of these runs, for visual clarity.

5. CONCLUSIONS

In this paper, we have introduced a novel method that uses physical layer characteristics of a wireless channel for generating a secret key between a pair of nodes in a wireless ad-hoc network.

Using the channel reciprocity and deep fades, our algorithms enable key agreement for a strong cryptographic key without the need of resorting to traditional key exchange cryptographic algorithms. The shared source of randomness between two nodes is the wireless channel which is unique to them. Given the lightweight computational requirements of our second procedure of section 3.2, it follows that relatively effortlessly the two wireless nodes can create a shared strong key that can be used for subsequent cryptographic operations.

We note that no special hardware is required for our techniques and a narrow-band filter along with a threshold detector are sufficient. The presence of a narrow-band filter before the threshold detector dramatically reduces levels of interference and noise for generating the bit vector. This provides robustness for different levels of SINR that permit communication between the two nodes. Our technique is also robust to channel estimation noise, since it is based on detecting deep fades, and not the complete channel impulse response which tolerates estimation errors, that may arise at the edges of deep fades and are shown to be correctable. Finally, in case the nodes move, their signal envelopes change which increases the entropy and can give rise to key generation at a quicker pace. If the nodes are stationery it may still be possible for the nodes to introduce interference on purpose so a key may be spawned. It should be stressed that security of our key generation mechanisms is not based on computational intractability assumptions such as those used to argue about security in schemes such as theÊDiffie Hellman key-exchange. For example, the key produced from our second procedure as detailed in section 3.2 is information theoretically secure for an adversary that is oblivious to the location of the deep fades (but still knows the number of them).

6. REFERENCES

[1] Y. Aumann, Y. Z. Ding, M. O. Rabin, Everlasting security in the bounded storage model, IEEE Transactions on Information Theory 48(6): 1668-1680 (2002)

[2] 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), pages: 3776-3784, Nov. 2005.

[3] R. Blom, "An optimal class of symmetric key generation systems," EUROCRYPT, pages: 335-338, 1984.

[4] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, "Perfectly-secure key distribution for dynamic conferences," Advances in Cryptology, pages:471-486, 1992.

[5] W.F. Bodtmann and H.W. Arnold, "Fade-Duration Statistics of a Rayleigh Distributed Wave," IEEE Transactions on Communications, vol. COM-30, No 3, p 549-553, Mar 1982.

[6] C. Cachin, U. M. Maurer, Unconditional Security Against Memory-Bounded Adversaries, CRYPTO 1997, pp. 292-306.

[7] S. A. Camtepe, B. Yener, Key Distribution Mechanisms for Wireless Sensor Networks: a Survey, TR-05-07 Rensselaer Polytechnic Institute, Computer Science Department, March 2005.

[8] S. A. Camtepe, B. Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks", in *Samarati et al (eds.), Computer Security-ESORICS,* Springer-Verlag, LNCS 3193, 2004.

[9] S. A. Camtepe, B. Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks", in *ACM/IEEE Transactions on Networking*, in press 2007.

[10] S. A. Camtepe, B. Yener and M. Yung, "Expander graph based key distribution mechanisms in wireless sensor networks," IEEE Int. Conf. on Commun., 2006.

[11] L. Carter and M. Wegman, Universal Hash Functions, J. Comp. and Syst. Sci. 18(2):143-154, 1979.

[12] H. Chan, A. Perrig and D. Song, "Random Key Predistribution Schemes for Sensor Networks," IEEE Symp. Security and Privacy, pages: 197, 2003.

[13] I. Csiszár and J. Körner. Broadcast channels with confidential messages. IEEE Transactions on Information Theory, 22(6):644-654, 1978.

[14] Y. Dodis, L. Reyzin and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In Advances in Cryptology Ů EUROCRYPT 2004.

[15] Y. Dodis and A. Smith. Entropic security and the encryption of high entropy messages. In J. Kilian, editor. First Theory of Cryptography Conference Ñ TCC 2005, volume 3378 of LNCS. Springer-Verlag, 2005.

[16] Y. Dodis, On Extractors, Error-Correction and Hiding All Partial Information, Information Theory Workshop (ITW), October 2005.

[17] Y. Dodis, L. Reyzin and A. Smith, Fuzzy Extractors, invited book chapter in "Security with Noisy Data", 2007.

[18] W. Diffie and M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory 22 (1976), pp: 644-654.

[19] B. Dutertre, S. Cheung and J. Levy, "Lightweight Key Management in Wireless Sensor Networks by Leveraging Initial Trust," System Design Laboratory, Technical Report, SRI-SDL-04-02, 2004.

[20] L. Eschenauer and V. D. Gligor, A key-management scheme for distributed sensor networks, ACM Conf. Computer and Commun. Security, pages: 41-47, 2002.

[21] J. E. Hershey, A. A. Hassan, and R. Yarlagadda, "Unconventional Cryptographic Keying Variable Management," *IEEE Transaction on Communications*, vol 43, No.1, pp3-6, January 1995.

[22] T. Holenstein and R. Renner, One-Way Secret-Key Agreement and Applications to Circuit Polarization and Immunization of Public-Key Encryption Advances in Cryptology - CRYPTO '05, Lecture Notes in Computer Science, Springer-Verlag, pp. 478-493.

[23] D.G. Howe, H. Hilden and E. Weldon, Jr., Shift correction code system for correcting additive errors and synchronization slips, United States Patent 5373513, 12/13/1994.

[24] J. Kittler and M. S. Nixon, Audio-and Video-Based Biometrie Person Authentication, 4th International Conference, AVBPA 2003, Guildford, UK, June 9-11, Lecture Notes in Computer Science, Vol. 2688, 2003.

[25] J.-P. M. G. Linnartz and P. Tuyls, New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates, 4th International Conference, AVBPA 2003, Guildford, UK, June 9-11, Lecture Notes in Computer Science, Vol. 2688, pp. 393-402.

[26] U. Maurer. Secret key agreement by public discussion. IEEE Transaction on Information Theory, 39(3):733-742, 1993.

[27] Ueli Maurer and Stefan Wolf. Unconditionally secure key agreement and the intrinsic conditional information. IEEE Transaction on Information Theory, 45(2):499-514, 1999

[28] A. Kitaura and H. Sasaoka, "A Scheme of Private Key Agreement Based on the Channel Characteristics in OFDM Land Mobile Radio," *Electronics and Communications in Japan, Part 3 (Fundamental Electronic Science*), vol 88, No 9, p 1-10, 2005

[29] B. Lai, S. Kim and I. Verbauwhede, "Scalable session key construction protocol for wireless sensor networks," IEEE Workshop on Large Scale Real-Time and Embedded Systems, 2002.

[30] Xiaohua Li, Mo Chen, and E. Paul Ratazzi, "Array-Transmission Based Physical-Layer Security Techniques For Wireless Sensor Networks," *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp 1618-1623, Niagara Falls, Canada, July 2005

[31] T. Ohira, "Secret Key Generation Exploiting Antenna Beam Steering and Wave Propagation Reciprocity," In: 2005 European Microwave Conference, vol(1), pages:9-12, Oct. 2005.

[32] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. Journal of Computer and System Sciences, 33:75–87, 1986.

[33] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC '89), pages 12-24. ACM Press, 1989.

[34] D. Stinson, Universal hash families and the leftover hash lemma, and applications to cryptography and computing, J. Combin. Math. Combin. Comput. vol.42, pp.3-31, 2002.

[35] Aaron D. Wyner. The wire-tap channel. Bell Systems Technical Journal, 54:1355-1387, 1975.