

Enhancing Wireless System Security with PHY-Layer Techniques

Alex Reznik, *Member, IEEE* and Yogendra Shah, *Member, IEEE*, Steven Goldberg, *Member, IEEE*
InterDigital, Inc
781 Third Ave
King of Prussia, PA 19406, USA
{Alex.Reznik, Yogendra.Shah, Steven.Goldberg}@InterDigital.com

Abstract — We present an approach to process channel observations in a wireless system and discuss how these may be used to enhance modern wireless security systems.

Index Terms — Perfect secrecy, RF channels, cross-layer security design

I. INTRODUCTION

A. A Brief Overview of Cryptography as it Stands Today

THE fundamental objectives of security technologies (or, more precisely cryptography) are usually identified as *confidentiality*, *integrity*, *authentication* and *non-repudiation* (see, e.g. [15]). These are defined as follows:

- *Confidentiality*: the message and any part of it are kept secret from all but the legitimate sender and receiver.
- *Integrity*: the receiver is able to make sure that the message has not been modified during transmission and a false message cannot be substituted for a real one.
- *Authentication*: the receiver should be able to verify the message origin.
- *Non-repudiation*: the sender should not be able to deny having sent the message at a later time.

Of these, the first three have been of primary concern in the design of security systems, although non-repudiation is gaining importance, for example with the advent of applications for digital signatures.

With one notable exception, all practical modern systems address these challenges using techniques which rely on computational cryptographic algorithms/ciphers. Since a cipher is the key element required for all aspects of security listed above, for most of this paper we shall concentrate on the process of ciphering and deciphering. The computational ciphers guarantee security and usability under the following assumptions:

- There is no possibility to break the algorithm using limited computational resources (usually in polynomial time).
- If a key is known, the deciphering can be accomplished relatively simply – or more precisely in polynomial time.
- The adversary’s computational power is limited: (s)he is unable to solve problems which are not in the computational class P.

These assumptions result in systems with some serious shortcomings. The need to decipher in polynomial time (in

computer science terms to verify in polynomial time) implies that typical ciphers belong to a set of problems in the computational class NP. While it is widely believed that such NP-complete ciphers cannot be broken in polynomial time, this has not been proven and this so-called P/NP equivalence is a key unsolved problem in computer science.

More seriously still, computational ciphers suffer from other shortcomings:

- Since the security is computation, these can be broken in principle.
- A lucky correct guess is usually easily verified.

These shortcomings notwithstanding, computational cryptography has been extremely effective in addressing many of the security challenges posed by modern communications and computer systems. This is done by introducing multiple layers of security making an attack on the overall system computationally (and, more importantly, economically) unjustifiable. A typical modern computational security system is organized according to the following guidelines:

- Authentication, if needed, is accomplished through pre-sharing a “strong secret” which is then used for authentication credentials.
- Public-key (PK) based techniques are used to protect authentication credentials when these are being exchanged.
- Actual ciphering is performed using symmetric keys. The keying data is derived from authentication credentials (the “strong secret”). When the strong secret is absent or cannot be used for some other reason, PK techniques can be used to exchange a key randomly generated by one party.
- Keys are frequently refreshed to maintain integrity of the system.

While quite good at addressing many of the security challenges posed by our inter-connected environment, these systems do display a number of weaknesses:

- Every exchange of data exposes the key data to potential attacks. Especially when keys are derived from authentication credentials (the identity), this is a serious problem. To address this, key hierarchies are used, which introduce several layers of protection between the actual keys in use (session keys) and the master key from which all key data is derived. However, maintaining a key hierarchy introduces its own overhead to the system operation.

- The time exposure to a successful attack of many systems is not bounded, i.e. a lucky attacker that manages to completely expose the key data can potentially eavesdrop on communications indefinitely. Key refresh procedures will not necessarily solve this problem, as many key refresh processes take as their starting point the previous key material.

Finally, while fairly robust at protecting data exchanged by communication networks, many such systems are not necessarily good at protecting the networks themselves. This is not an issue when the network is either essentially irrelevant, as is the case with the Internet which only cares about end-to-end data exchange and application interface; or when the network is inherently secure as is the case with wired LAN where access to the network is usually symptomatic of a serious security breach in the physical (as opposed to the virtual world) and should be addressed as such.

However, as wireless networks emerge as a key asset for enterprises large and small, it is becoming clear that data-centric approaches to security leave the viability of the network itself exposed. Because wireless networks do not enjoy the physical protection of wires, their vulnerability presents an appealing target. Even if an attacker cannot access the data, frequently he can create tremendous damage by just bringing down the network. Worse yet, the weakness in the access layer of a wireless network may present a back door to attacks that eventually lead to compromise of the key data.

We therefore argue that lower-layer (including Physical or PHY layer) security techniques must be a part of an integrated security solution in wireless networks. Additionally, we demonstrate that when properly utilized these can be used to enhance the performance of higher layer security techniques while closing the security gaps at the lower layers.

In what follows we discuss one such technique which accomplishes two practical goals:

- It provides security (more specifically confidentiality) associated with a specific physical link in a wireless network. Thus, it makes the link more akin to a wire link which is inherently secure against any but a true physical attack (the placement of a tap on the wire).
- It alleviates the time exposure of the higher layer security systems by strictly limiting it in both the forward and backward direction.
- It separates the connection between confidentiality and authentication, often an unavoidable byproduct of traditional security procedures.

As we shall see, it also has the more theoretical advantage of not allowing a lucky attacker to easily verify his correct guess.

We also briefly discuss some possible approaches for combining this technique into a holistic security solution for a wireless network. This work combines our proposed approach with other techniques, such as the authentication approaches of [11]-[14] and points to a roadmap for building a complete secure wireless system.

B. Information-theoretic Cryptography

Before we proceed with our discussion we need to overview a different approach to cryptography which relies on information theoretic guarantees of security that are not subject to computational assumptions.

Suppose that Alice and Bob are two parties who respectively observe two correlated sources of randomness (In Figure 1 the reciprocal channel \mathbf{H}). They wish to generate a common secret key by communicating over a noiseless public channel. The secret key generated by Alice and Bob should be effectively concealed from eavesdropper Eve who observes the transmissions on the public channel.

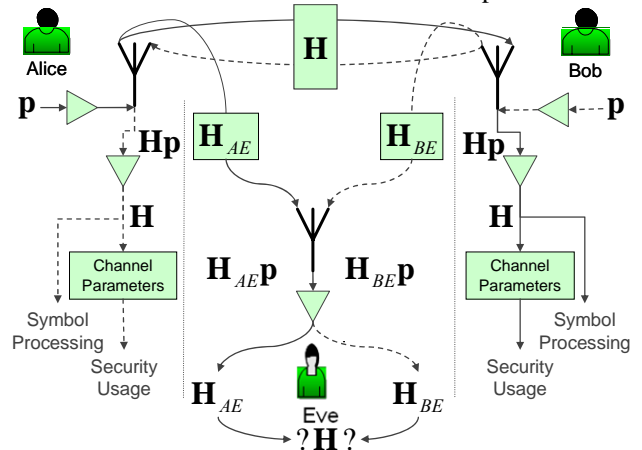


Figure 1. Communicating Parties and Eavesdropper Communication Paths

It is also assumed that the noiseless public channel, over which Alice and Bob can communicate with each other, has been authenticated, i.e., Eve is passive or unable to tamper with the transmissions on the public channel without being noticed by Alice or Bob. Much work [1]-[5] has been devoted to the study of secret key generation for this “passive eavesdropping” case. The case where Eve is an active adversary has been studied in [2] where it is shown that the underlying techniques reduce to first authenticating the channels and then using the techniques appropriate for an authenticated channel. While the work in [1]-[5] is discussed in the context of discrete secrecy sources, extensions to continuous sources are straightforward [8].

While in the most general case Eve (the eavesdropper) may have access to useful side information, we shall assume that this is not so and hence the only useful information Eve has is obtained from observing the communication over the public channel (In Figure 1 Eve’s channels with Alice and Bob are \mathbf{H}_{AE} and \mathbf{H}_{BE}). As we shall see, this assumption is well justified by the wireless channel properties that we shall come to rely on. Furthermore, it greatly simplifies our discussion.

The secret key is defined as follows: Let $\mathbf{X}=(X_1, \dots, X_n)$ and $\mathbf{Y}=(Y_1, \dots, Y_n)$ be n independent and identically distributed repetitions of the correlated random variables X and Y . Alice and Bob respectively observe the sequences \mathbf{X} and \mathbf{Y} . Furthermore, they can communicate with each other over a noiseless public channel, possibly interactively in many rounds. Let \mathbf{V} denote all the transmissions on the public channel. After the transmissions, Alice can generate a bit string S_A , based on (\mathbf{X}, \mathbf{V}) , and Bob can generate a bit string S_B , based on (\mathbf{Y}, \mathbf{V}) . A bit string S constitutes a secret

key for Alice and Bob if there exist S_A and S_B such that

$$\Pr(S = S_A = S_B) \approx 1 \quad (1)$$

$$I(S; V) \approx 0 \quad (2)$$

$$H(S) \approx |S| \quad (3)$$

where $|S|$ denotes the length of the bit string S ; I denotes mutual information between two random variables and H denotes the entropy of a discrete random variable. (e.g. [16] for background on information measures.) The conditions (1-3) are interpreted as follows: Alice and Bob generate “almost” the same secret key (from (1)); this secret key is nearly uniformly distributed (from (3)); and this secret key is nearly “statistically independent” of Eve’s information, i.e., the transmissions V on the public channel (from (2)). Hence, this secret key is effectively concealed from Eve. Note that this concealment is stronger than computational secrecy in two key features: it is not predicated on any computational assumptions; a correct guess is not verifiable – even if Eve happens to guess the correct key, she has no means of verifying that she has done so.

The (entropy) rate of a secret key, viz. $\frac{H(S)}{n}$, is called a secret key rate. The largest secret key rate is called the secret key capacity, denoted by C_s . The notion of secret key capacity indicates the length of the longest secret key that can be generated by Alice and Bob, based on their observations \mathbf{X} and \mathbf{Y} .

It has been shown [1],[3] that the secret key capacity for the model above is:

$$C_s = I(X; Y) \quad (4)$$

Further, it is known [1],[3] that the secret key capacity can be achieved by a single transmission from Alice to Bob (or, vice versa).¹ The process may generally be described as follows²:

- Alice and Bob each observe their sources.
- Alice knows that Bob observed something close to what she has observed but with some errors. She pretends that this was accomplished via a secret side channel transmission that introduced errors (note no actual transmission has occurred). To correct these errors she needs to forward some error correction information over the public channel – and she does so.
- Once Bob receives the error correction information, he can correct his “errors” – i.e. adjust his data to be identical with that of Alice.
- At this point both Alice and Bob possess a common string of data. However some of that information has

been leaked via the public transmission of information. This can be removed, by, for example, hashing (the privacy amplification step of [1]).

- Because the random source is constantly renewing itself, the process is repeated at regular intervals resulting in a positive secrecy rate.

At this point, we should ask ourselves the following questions: is this theory realizable in practice? If so, why isn’t it widely used? And finally at this point one has only generated a secret key – while this key is “perfectly secret,” how can it actually be used to protect information in a way that is better than the existing methods? As we shall see the answers to the latter two questions highlight the key limitations of this approach which our wireless channel based cryptography will have to address.

We begin, however, by pointing out that information theoretic cryptography has been successfully implemented in practice. Quantum cryptography systems (which are now being tested across the world) rely on information-theoretic cryptography as opposed to computational cryptography techniques. They use quantum entanglement as the source of correlated randomness and operate along the lines discussed above. See [17] for more information on quantum cryptography.

This begs the question of why isn’t information-theoretic cryptography widely used. The reason is that we are effectively trading one severe limitation (the computation assumption) for another – the need for Alice and Bob to possess *a priori* correlated source – i.e. they must possess a secret to begin with (and, as we shall see shortly, the amount of secret information must be effectively infinite). We simply allow for this secret to be “dirty” (i.e. statistically dependent but not identical sources) and go through a procedure to clean it up. It can be easily shown (see, e.g. the discussion in [2]) that absent such an *a priori* “dirty secret” no information-theoretic key agreement is possible.

The issues get ever more severe when one considers what should be done with the secret keys when these are generated. A natural approach was taken by Shannon in one of the first theoretical treatments of cryptography [18]. He imposed the strict condition that the mutual information between the message (M) and the ciphertext (C) should be zero, i.e. absent the key, the ciphertext should reveal nothing about the message. In what we shall refer to as Shannon’s Pessimistic Result (SPR), he showed that this could only be accomplished if the entropy of the key (K) is at least as large as the entropy of the message:

$$H(K) \geq H(M) \quad (5)$$

If the key and the message are treated as i.i.d bit strings where each bit is either 0 or 1 with probability $\frac{1}{2}$, then SPR reduces to the requirement that we have a *new bit of key* for each new bit of message. Incidentally, if we do have a new bit of key for each new bit of message, a simple bit-wise XOR is easily shown to be a cipher that results in zero mutual information between the message and the ciphertext. Thus, this cipher, referred to as the *one-time pad* (or the

¹ We note that since the transmission result is highly dependent on the assumption that Eve has no useful side information. When this is not so, the algorithms described here may still be used with some modification; however they yield performance that is far from optimal. To optimize performance an additional, highly interactive step is needed before the algorithms described here are used. See [1] and [2] for more detail.

² The procedure is closely related to the problem of distributed source coding. e.g. see the material on Wolf-Slepian coding in [16], Wyner-Ziv coding in [20], and also discussion in [8].

Vernam cipher) is optimal in the Shannon sense. Unfortunately, if (5) is not satisfied (i.e. key bits have to be re-used), the one-time pad is quite bad.

Because of SPR secret key rate and secrecy rate are sometimes equated in the information-theoretic literature. However, as we shall see such an equivocation would lead to extremely low secrecy rates in our application (thus the “pessimistic” in SPR). Therefore, one of the questions that we shall have to answer is whether and how a low, but positive secret key generation mechanism can be put to use in a wireless communication system.

II. SECRECY GENERATION FROM WIRELESS CHANNELS

Suppose that a pair of wireless terminals communicate with each other on the same frequency in a wireless communication environment. The mutual wireless channel between these two terminals can serve as a common random source:

- The commonality follows from the reciprocity of the wireless channel.
- The randomness is well known to be inherent in wireless channels due to the constantly changing physical environment.

In most cellular channels it is further enhanced by the high number of scatterers. If both terminals possess some means of observing their mutual channel, the resulting observations are highly statistically dependent. On the other hand, the observations of a third terminal almost certainly remain independent of the channel-specific observations of the first two terminals if the third terminal is located more than half a wavelength away from these two terminals.

The use of the wireless channel for secrecy has received some attention from the research community over the last few years. There has been some early work ([6],[7]) however, there was little immediate follow-up to this work. We suspect one of the reasons for the lack of follow-up to this work is that it suffers from the SPR problem. The data is modulated directly onto the secret channel to obtain secrecy – and thus either the rate needs to be limited by the secrecy rates of wireless channels (which, as we shall see are very low), or the actual security delivered is questionable.

Recently, there have been some efforts to apply Wyner’s wire-tap model to fading channels [21]-[24]. However these approaches are either subject to assumptions that the propagation conditions to the eavesdropper are known (which is not reasonable); or are limited to eavesdroppers which are sufficiently far away to satisfy some non-negligible assumption on the channel quality (where the distance is much greater than the fraction of the wavelength which we shall assume). While these approaches have merits and potential applications, we propose an approach that avoids the limitations inherent here.

None of the efforts above attempt to exploit a wireless channel as a source of shared and secret randomness for explicit secrecy extraction. However, to do so presents perhaps the most practical approach as the extracted secrecy is available for any use, as opposed to just being restricted to some specific contexts, like the approaches taken in the

references above. One approach to the problem has been studied previously in the context of UWB channels [10]. In this paper we shall concentrate more on challenges associated with channels of much narrower bandwidth than UWB channels, such as those commonly observed in cellular and wireless LAN communications; however UWB channels are an important area of potential application for these techniques.

Whatever wireless system a key generation is attempted in, the process is effectively partitioned into the following steps:

- i.) A channel impulse response (CIR) is obtained by the two key generating parties, Alice and Bob.
- ii.) The CIR is processed to generate an appropriate representation (source coding of CIRs).
- iii.) Once a sufficiently large block of quantized CIRs is aggregated, a block of secret bits (without loss of generality we assume bit generation) is generated through interactive communication as outlined above.

Each of these steps presents its own unique challenges which we discuss next.

A. CIR Generation

Of the three steps described above CIR generation is perhaps the most straightforward. In principle, any standard existing channel estimation technique may be used by Alice and Bob independently. In most systems these typically utilize pilots/training sequences although this is not required. The key challenge here is to make sure that Alice and Bob estimate the *same* time epoch of the channel. This is not hard in typical channels which change very slowly. The coherence times for pedestrian channels are in the hundreds of milliseconds and even larger for more static channels. Even for vehicular channels with speeds as high as 250 km/h, coherence times remain on the order of milliseconds – sufficiently long to be able to synchronize measurements (and if necessary a packet exchange to facilitate these) in most modern systems.

B. Source Coding of CIRs

A much more difficult issue is the processing of the observed CIR to produce a representation appropriate for secrecy generation. Recall that we require our final result (the key) to i) be identical at both Alice and Bob and ii) to have maximal entropy – i.e. be a bit string with independent bits with probability distribution $[\frac{1}{2}, \frac{1}{2}]$. Furthermore, we wish to generate as long a key as possible – and thus to minimize the information leaked during the key agreement phase of the protocol.

Unfortunately, the CIR measurement of a multipath or dispersive channel typically consists of highly correlated data. We are thus faced with one of two choices. We can proceed with this data as is, generate a key which is the same on both sides but is not fully compressed and then run a compression algorithm on it. Alternatively, we can first compress the measured CIRs on both sides and then run the key agreement algorithm on what is now fully compressed data resulting in a fully compressed key.

The former approach (agree first, compress second) requires an error-correction code capable of correcting

errors in highly correlated data. Such codes do not exist in practice (although information theoretically it is known how to do this). Conversely, using a standard code (which treats all data as independent/fully compressed) would result in too many error-correction bits being exchanged to perform key agreement, reducing the resulting rate. For wireless channels in the cellular and WLAN range, the reduction would effectively drive the secrecy rate to zero.

The latter approach (compress first, agree second) avoids this problem, provided that a compression algorithm is used which produces similar outputs for similar inputs. Unfortunately, many compression algorithms also act as excellent hashes with the result that two strings which start out similar get compressed into very distinct bit strings. Thus, standard compression techniques cannot be applied.

The problem may be addressed in one of several ways and we concentrate on the approach proposed in [9]. Specifically, we start with the assumption that the channel results from superposition of signals generated by a relatively low number of rather discrete scatterers, and decompose the CIRs back into discrete multipath components. In doing so, we must note the following:

- We do not insist on correctness in the sense that we don't care whether the resulting components correspond to the actual constituent multipath components of the channel.
- We require independence – i.e. our solution must provide assurance of independence of outputs, at least under well-defined and reasonable assumptions.
- We require consistency. Small variations in the starting CIR should result in highly similar (statistically dependent decompositions).

The reference [9] proposes one approach, based on a greedy algorithm, which satisfies all these constraints. It should be noted that because of the underlying assumption of low number of discrete scatterers, this approach is best utilized for cellular and WLAN systems. A different approach is likely to be more effective for UWB systems (see, e.g. [10]).

C. Key Agreement

Once a CIR is processed into an appropriate form, key agreement must take place. In the case of wireless channels, the sources are inherently continuous while the end result (the key) is inherently discrete. This presents an additional challenge to be dealt with.

The case of continuous sources, in the special case where these are Gaussian, has been addressed in [8]. The algorithms proposed in [8] use well known LDPC codes and achieve secrecy rates within about 1 bit of capacity, as illustrated in Figure 2. As Figure 2 demonstrates, soft decision and gray coding are critical in obtaining good performance. Further improvements may be obtained through the use of vector quantization [25]. However as the gain is bounded by about 1 bit per path per channel realization, the gains do not necessarily justify the significant added complexity.

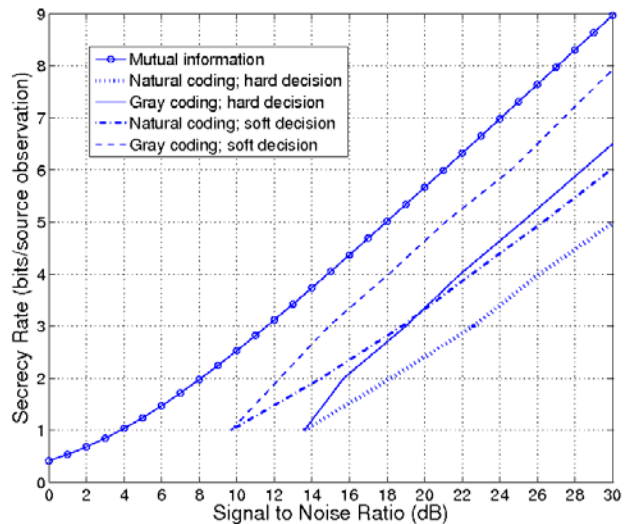


Figure 2. Key Rates Achievable for Gaussian (Rayleigh) Fading using Techniques in [8]

Reference [9] also uses these techniques to produce the following results for several ITU channel models as shown in Figures 3 and 4. We note that the SNRs shown here are for the channel estimate after the channel estimation process. Since training-sequence (or pilot) based channel estimation typically carries significant processing gains, the SNRs shown here are the reasonable ranges which one may hope to achieve.

D. The Actual Key Rates and System Challenges

The results above presented key rates in terms of the number of bits per independent channel realization. To translate these into actual secret key rates we must take into account the coherence times of wireless channels, as these are roughly the time intervals at which the channels renew themselves. The coherence time of a wireless fading channel is given by [26]

$$T_c \approx \frac{9\lambda}{16\pi V} \quad (6)$$

where V is the velocity in meters per second and λ is the wavelength of the carrier frequency. At a carrier frequency of 2GHz, $\lambda \approx 0.15$ meters, and at velocities of 3km/h, 30km/h and 120km/h, we get coherence times of 32mSec, 3.2mSec and 0.8mSec respectively. Thus, for example for the ITU PB channel (see Figure 3) at SNR of 10 dB and velocity of 3km/h we can obtain roughly $(1/0.032) \approx 31$ secret bits per second (sbps). For the same channel and velocity, but at SNR of 30dB, we can obtain well over 900 sbps. Similarly, for the VA channel (see Figure 4) at SNR of 20dB we can obtain about 3,400sbps at 30km/h and 13,600sbps at 120km/h.

Recalling SPR, we note that given these key rates information-theoretic “optimal” encryption is not feasible as the data rates would be limited by the secret key rates. On the other hand, we note that we are now in possession of a new resource – perfect secrecy – and a resource which is quite rare in nature (other than wireless channels, only quantum entanglement has so far been found to display the necessary properties). We are then left to wonder how we take advantage of this rare natural resource.

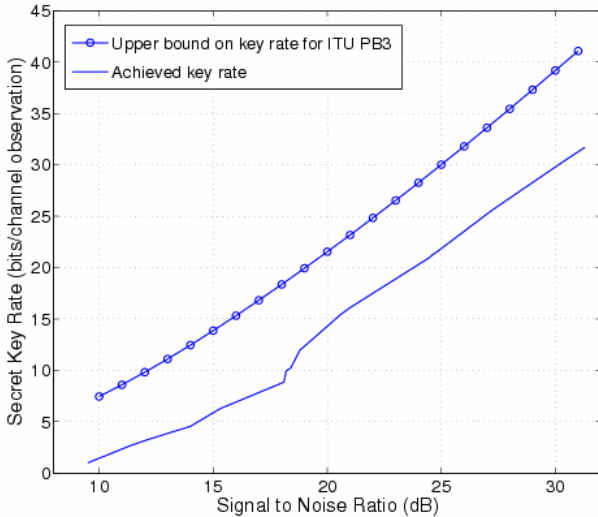


Figure 3. Achieved Secrecy Rates for ITU PB and Upper Bound

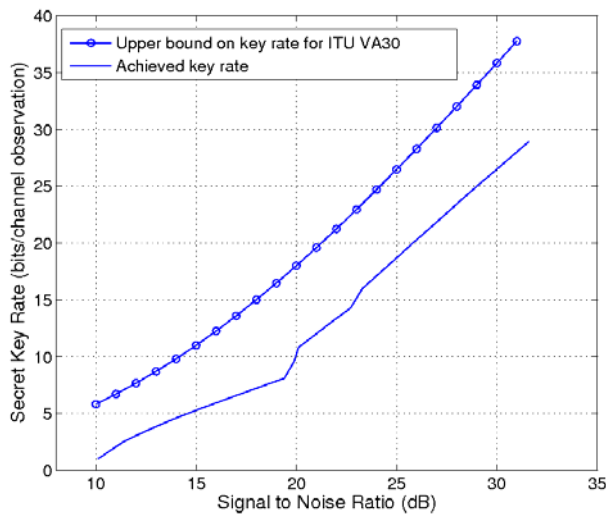


Figure 4. Achieved Secrecy Rates for ITU VA Model and Upper Bound

This brings us full circle to the central question raised in the introduction – how can physical layer techniques be incorporated into holistic security solutions. Having demonstrated the possibilities which exist at the physical layer in wireless systems, we now return to the system issue.

III. CROSS-LAYER SECURITY IN WIRELESS SYSTEMS

In this section we explore how this precious source of secrecy, the Information-Theoretic Secure (ITS) bits described previously may be used to improve existing wireless security systems.

Moreover, because a powerful error-correcting code (such as LDPC) must be utilized there may be a significant delay (especially under quasi-static channel conditions) after communication starts before the ITS bits actually become available. Clearly, this resource is invaluable; the question, however, is how to utilize it to its fullest potential given its limited availability. The key, we believe, is to use this resource to secure the very heart of the encryption capabilities of wireless systems, and to take advantage of its renewable nature to ensure absolutely secure key refresh.

Although these basic concepts may be applied to any wireless system, by way of example, we demonstrate how the existing 802.11i security protocol may be improved.

Among the innovations of 802.11i is the introduction of two authentication modes. The first mode is based on the availability of an authentication server (hereafter called 802.1x-based authentication). The second mode is based on configuring a secret password or pass-phrase on the participating devices i.e. the Pre-Shared Key mode (PSK). It should be noted that, with PSK mode, the implicit assumption is that a user authenticates by demonstrating knowledge of the secret key.

Although the existing 802.11i protocol features both very robust and secure encryption algorithms and an automated key distribution framework, it cannot address certain security threats intrinsic to conventional security design. In fact, one of the easiest ways to attack a secured system is to gain unauthorized access to the credentials of a legitimate user. For example, suppose a malicious attacker is able to observe (or obtain knowledge of) the pass-phrase employed by a user when configuring his home WLAN or when accessing a public hotspot, the attacker will then have all the information he needs to break the encryption cipher of the legitimate user. Thus the attacker may gain access to the WLAN and also eavesdrop on any traffic the legitimate user transmits.

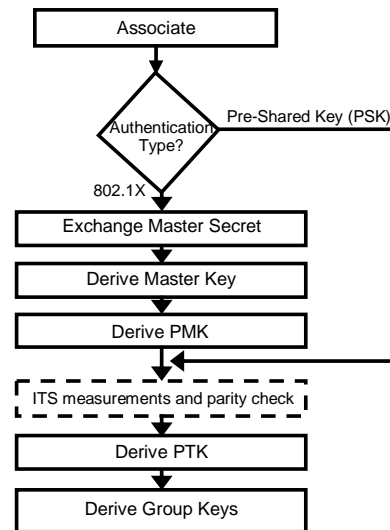


Figure 5. Using ITS in 802.11i Security

To demonstrate how the introduction of ITS bits can improve the 802.11i protocol, we begin with minimal modification. In Figure 5 we show how a WLAN transmission using PSK authentication mode can be modified using channel-based secrets.

In this approach, we simply use ITS strings obtained in Section II to derive the Pairwise Transient Key (PTK) from the PMK, $PTK = PRF \{PMK, \text{Info in the clear}, ITS\}$.

The measurements required to generate the ITS bits can be carried out at any time prior to deriving the PTK (not just after deriving the PMK as shown in Figure 5). Likewise, the parity check exchange can also be carried out at any time prior to deriving the PTK. Once the ITS bits and PMK are derived (in case of 802.1x) the PTK can be derived from the PMK using a pseudo-random function as shown above. The Group Key derivation and distribution can be left untouched.

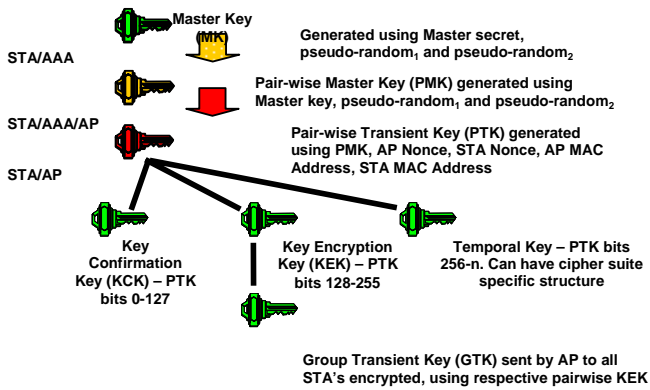


Figure 6. 802.11i Security Key Hierarchy

Having illustrated a simple way to make good use of the ITS bits available to us, we now depart from the key hierarchy in Figure 6 to show how we might maximize the potential benefits of the ITS. We have only one key. If 802.1x authentication is used, the AAA server provides the STA its credentials. The STA then verifies the AAA servers' credentials and provides its own credentials along with a secret. The AAA server then forwards the secret to the AP after verifying STA credentials. An Encryption Key (EK) is then derived by the STA and AP, using the secret and the ITS string. If authentication is PSK, then the Pre-Shared Key acts as the secret. Part of the EK is used for verification, while part of it is used to protect group keys derived later. The remaining part is the portion actually used in the AES algorithm. We note that the key hierarchy in the proposed scheme is significantly simpler than the one in Figure 6. In fact, the hierarchy no longer exists. Instead, we simply have the following two sets of keys:

- The Pre-Shared Secret used for authentication
- An intermittently updated Encryption Key which is used for actual data transmission.

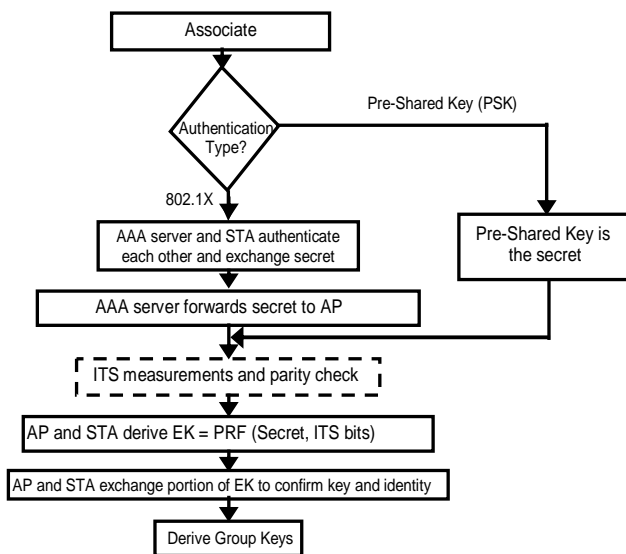


Figure 7. New 802.11i Key Derivation Protocol

Let us examine how the proposed scheme addresses the security threats we discussed above. Suppose that an eavesdropping terminal has been able to gain access to a user's PSK and is eavesdropping on the transmission. With

the scheme in Figure 6, the eavesdropper is able to obtain the PMK – but can go no further. Unless the eavesdropper is physically co-located with either the AP or the terminal, it cannot obtain the ITS bits used to derive the PTK.

Similarly, the scheme in Figure 7 permits the eavesdropper to realize that authentication has transpired, but does not allow eavesdropping on actual data being transmitted.

Note also that the proposed scheme provides specific forward and backward security. Since ITS strings are constantly generated (we have a positive ITS rate over time), this fresh set of ITS bits can then be immediately used to derive a new PTK or a new EK for communication. Since the terminals involved generate the ITS bits in sync, the process is naturally synchronous. Forward and backward secrecy versus information transmitted with previous PTK/EK is immediately achieved: even if an eavesdropper obtains the current PTK/EK, it is statistically uncorrelated to previous or to future ones, and therefore the eavesdropper is unable in principle to learn those keys. Thus, even when the key is completely exposed, data is vulnerable only during the period of time it takes to accumulate enough ITS bits for a new key.

Finally, let us consider the case of a determined and computationally powerful attacker whose aim is to actually obtain the PSK. A simple modification of the protocols presented prevents the attacker from being able to work back towards the PSK — even if the attacker obtains a particular set of ITS bits. The required change is simply to not use the PSK in the derivation of the PTK/EK. The gain here is a complete separation of the authentication procedures from the data encryption processes.

Now that we have considered how this source of physical layer information may be used to improve existing wireless security let us consider what other uses this information may be put to use. A general weakness of wireless security is Denial of Service (DoS) attacks. For example consider a scenario where the MAC layer is receiving correctly received packets from the physical layer. However, the rate of reception is somewhat lower than the apparent physical channel capacity. At the MAC layer it is difficult to separate this problem between that of a subtle DoS attack or genuine poor channel quality. However, recall that the original premise of the ITS security is that the channel observations between two nodes communicating with each other are mutually unique and significantly different from a third eavesdropping node as long as that node is at least half a wavelength away. This characteristic enables the terminals to in essence consider the CIR as a finger-print of the transmitter (see [13]) from which the signal is emanating. Hence, if during the authentication process, the receiving terminal measures the CIR and then tracks this CIR during subsequent transmissions then in principal if an adversary were to transmit bogus messages in order to launch a DoS attack, then the fact that the signature or finger-print of the received signal differs from the legitimate signature measured during authentication and subsequently tracked, would suggest that the packets being received are not from the same node. In essence using the physical layer information at our disposal, we are able to continuously authenticate packets at the physical layer. This form of security is only possible through physical layer security mechanisms. Other physical layer security enhancements

are being explored in the research community [12] and show promise in providing overall robust cross-layer wireless security.

IV. SUMMARY AND CONCLUDING REMARKS

In this paper, we discussed the potential of using physical layer properties of a wireless channel to reinforce wireless security systems. We demonstrated how wireless channel reciprocity may be used to extract secret keys; and how these, in turn, may be used to enhance the existing 802.11i protocol. Additional exploitation of the physical layer properties can help towards achieving a more robust wireless security. More importantly, these modifications close security gaps that have until now remained open to attackers.

The authors wish to thank Rajat Mukherjee and Akbar Rahman for their contribution on the 802.11i protocol analysis and enhancements discussed in this paper.

REFERENCES

- [1] U. Maurer, "Secret key agreement by public discussion," *IEEE Trans. on Information Theory*, vol. IT-39, pp. 733–742, 1993.
- [2] U. Maurer and S. Wolf. Secret key agreement over a non-authenticated channel — parts I-III: Definitions and bounds. *IEEE Transactions on Information Theory*, IT-49: 2003.
- [3] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography — part I: Secret sharing," *IEEE Trans. Information Theory*, vol. IT-39, pp. 1121–1132, 1993.
- [4] I. Csiszar and P. Narayan. Common randomness and secret key generation with a helper. *IEEE Transactions on Information Theory*, IT-46:344–366, 2000.
- [5] I. Csiszar and P. Narayan. Secrecy capacities for multiple terminals. *IEEE Transactions on Information Theory*, IT-50:3047–3061, 2004.
- [6] A. A. Hassan, W. E. Stark, J. E. Hershey and S. Chennakeshu, "Cryptographic key agreement for mobile radio," *IEEE Digital Signal Processing Magazine*, vol. 6, pp. 207–212, 1996.
- [7] H. Kooraparty, A. A. Hassan and S. Chennakeshu, "Secure information transmission for mobile radio," *IEEE Comm. Let.*, vol. 4, pp. 52–55, 2000.
- [8] C. Ye, A. Reznik and Y. Shah, "Extracting Secrecy from Jointly Gaussian Random Variables," *Proc. International Symposium on Information Theory*, pp. 2593-2597, 2006.
- [9] C. Ye, A. Reznik, G. Sternberg, Y. Shah., "On the secrecy capacities of ITU channels," to appear in *Proc. VTC Fall 2007*.
- [10] R. Wilson, D. Tse and R. A. Scholtz, "Channel Identification: Secret Sharing Using Reciprocity in Ultrawideband Channels," submitted to *IEEE Trans. Information Forensics and Security*, 2006.
- [11] Q. Li and W. Trappe, "Reducing Delay and Enhancing DoS Resistance in Multicast Authentication through Multi-grade Security," *IEEE Trans. On Information Forensics and Security*, vol.1, pp. 190-204, 2006.
- [12] W. Xu, K. Ma, W. Trappe, Y. Zhang, "Jamming Sensor Networks: Attack and Defense Strategies," *IEEE Networks Special Issue on Sensor Networks*, vol. 20, pp. 41-47, 2006.
- [13] L. Xiao, L. Greenstein, N. Mandayam, W. Trappe, "Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication," *Proc. ICC 2007*, pp. 4646-4651.
- [14] Q. Li, W. Trappe, "Lightweight detection of Spoofing Attacks in Wireless Networks," *Proc. IEEE. Int. Conf. on Mobile Adhoc and Sensor Systems (MASS)*, 2006, pp. 845-851.
- [15] H. Delfs and H. Knebl, *Introduction to Cryptography. Principles and Applications.*, Springer, New York, 2002.
- [16] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.
- [17] M. A. Nielsen and I. L. Chung, *Quantum Information and Quantum Computation*, Cambridge U. Press, New York, 2000.
- [18] C. E. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*, vol. 28, pp. 656–715, 1949.
- [19] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *J. Amer. Inst. Elect. Eng.*, vol. 55, pp. 109–115, 1926.
- [20] S. Shamai(Shitz), S. Verdú, and R. Zamir, "Systematic lossy source channel coding," *IEEE Transactions on Information Theory*, vol. IT-44:564–579, 1998.
- [21] J. Barros and M. R. D. Rodrigues, "Secrecy Capacity of Wireless Channels," *Proc. IEEE Int. Symp. on Inform. Theory*, pp. 356–360, 2006.
- [22] M. Bloch, J. Barros, M. R. D. Rodrigues and S. W. McLaughlin, "Information-Theoretic Security for Wireless Channels: Theory and Practice," *Proc. Inform. Theory and Applications Workshop*, San Diego, USA, Feb. 2007.
- [23] Z. Li, R. Yates and W. Trappe "Secret Communication with a Secret Eavesdropper Channel," *Proc. IEEE Int. Symp. on Inform. Theory*, pp. 1296-1300, 2007.
- [24] P. K. Gopala, L. Lai and H. El Gamal, "On the Secrecy Capacity of Fading Channels" *Proc. IEEE Int. Symp. on Inform. Theory*, pp. 1306-1310, 2007.
- [25] G. Van Assche, J. Cardinal and N. J. Cerf, "Reconciliation of a Quantum-Distributed Gaussian Key," *IEEE Transactions on Information Theory*, IT-50:394–400, 2004.
- [26] R. Steele, *Mobile Radio Communications*, (reprint), Wiley, New York, 1996.