Signal-Flow-Based Analysis of Wireless Security Protocols

Çağatay Çapar[a], Dennis Goeckel[a], Kenneth G. Paterson[b,*], Elizabeth A. Quaglia[b], Don Towsley[a], Murtaza Zafer[c]

[a] *University of Massachusetts*
[b] *Royal Holloway, University of London*
[c] *IBM T.J. Watson Research*

Abstract

Security protocols operating over wireless channels can incur significant communication costs (e.g., energy, delay), especially under adversarial attacks unique to the wireless environment such as signal jamming, fake signal transmission, etc. Since wireless devices are resource constrained, it is important to optimize security protocols for wireless environments by taking into account their communication costs. Towards this goal, we first present a novel application of a signal-flow-based approach to analyze the communication costs of security protocols in the presence of adversaries. Our approach models a protocol run as a dynamic probabilistic system and then utilizes Linear System theory to evaluate the moment generating function of the end-to-end cost. Applying this technique to the problem of secret key exchange over a wireless channel, we quantify the efficiency of existing families of key exchange cryptographic protocols, showing, for example, that an ID-based approach can offer an almost 10-fold improvement in energy consumption when compared to a traditional PKI-based protocol. We then present a new key exchange protocol that combines traditional cryptographic methods with physical-layer techniques, including the use of "ephemeral" spreading codes, cooperative jamming, and role-switching. Utilizing signal flow analysis, we demonstrate that this new protocol offers performance advantages over traditional designs.

Keywords:
Security protocols, wireless, cost, Linear System, Physical layer, key exchange

*Corresponding author
Email addresses: ccapar@ecs.umass.edu (Çağatay Çapar), goeckel@ecs.umass.edu (Dennis Goeckel), kenny.paterson@rhul.ac.uk (Kenneth G. Paterson), Elizabeth.Quaglia.2008@live.rhul.ac.uk (Elizabeth A. Quaglia), towsley@cs.umass.edu (Don Towsley), mzafer@us.ibm.com (Murtaza Zafer)

1. Introduction

With the wide proliferation of wireless devices, securing information sent over wireless channels is imperative and has rightfully received significant research attention. However, the focus of the design and evaluation of security protocols for wireless environments has been on traditional metrics of security, such as resilience against various attacks, key-sizes, computational complexity tradeoffs, etc. While these are important metrics, it is also equally important to consider the communication cost (energy, delay) of a security protocol, especially since wireless devices generally operate under strict resource constraints such as limited battery energy.

Inherently, communications over wireless channels is probabilistic in nature due to random errors caused by signal fading, shadowing, and noise, and due to potential adversarial attacks such as signal jamming, fake signal transmission, etc. Therefore, evaluating the end-to-end performance of a security protocol becomes especially difficult when considering a wireless setting. For example, suppose that a security protocol requires the exchange of certain messages over an open wireless channel that is subject to adversarial jamming. Since a message transmission can fail or the message can be corrupted, the protocol would in reality undergo multiple re-trials to deliver each message and/or require re-starts from different points. As the logical protocol flow is probabilistic, evaluating the end-to-end cost is non-trivial. Furthermore, depending on the protocol design, an attacker may be able to exploit the need for such multiple trials and re-starts and force a significant communication cost. Thus, though a cryptographic protocol may provide strong security guarantees, it may also be very inefficient in terms of average resource consumption when communication costs are accounted for. We therefore argue that an *important tradeoff for wireless security protocols is their efficiency measured in terms of the communication cost incurred versus the level of security achieved.*

In this paper, we utilize Linear System theory to develop a signal-flow-based approach to analyze wireless security protocols. The main idea here is to transform a protocol flow chart into a signal flow graph (by assigning probabilities and costs on individual branches) and then utilize reduction techniques to deduce the end-to-end transfer function, from which the end-to-end costs of the protocol can be computed. To concretely present this approach, we consider as a running example the fundamental problem of *secret key exchange over an open wireless channel in the presence of an active adversary.* This will provide an underlying context throughout the paper. We describe this problem next.

1.1. Wireless Key Establishment

Bootstrapping security over a wireless channel requires first establishing a jam-resilient communication channel, since otherwise open air transmissions are highly susceptible to disruption attacks such as signal jamming. An approach generally employed in this setting is to use spread-spectrum communications (e.g. frequency-hopping spread spectrum (FHSS)), which limits an attacker's ability to jam the communication signals without expending large amounts of

energy [1]. However, establishing a spread-spectrum channel requires the participating parties to either already pre-share or securely establish a cryptographic key, enabling them to select a 'private' spread spectrum channel that is unknown to the attacker. In turn, this requires any pair of network nodes that might wish to establish jam-resilient wireless communication either to have available a pre-established key, or to run a key establishment protocol over an 'open' wireless channel prior to switching to a spread spectrum channel determined by the agreed key.

Consider first the case of using pre-established keys. If we consider the setting where we have a large number of network nodes that may wish to establish secure communications and where node compromise is a realistic threat – for example in a military environment or an emergency scenario – then having a single, system-wide pre-established key is not a viable solution, since compromise of a single node then compromises the whole network. On the other hand, having a unique pre-shared key per possible pair of communicating parties is not a good solution either, since it does not scale well and is inflexible once deployed. Intermediate solutions, such as those proposed in [2, 3], scale better, but may still require substantial key storage at the nodes. Instead, establishing shared secret keys on-demand by utilizing cryptographic protocols over the open air is a more flexible approach and may, in fact, be necessary in many emergency and military scenarios.

There is a large body of research on cryptographic protocols for key exchange from public messages [4], but when employed over wireless channels the messages exchanged during key establishment are subject to active adversarial attacks. Because of adversarial attacks (as well as the inherently noisy communications environment), the protocol participants may be forced to repeat steps, or even re-start the protocol from scratch, many times before a session key is successfully established. This implies that establishing a private spread spectrum communications channel may incur significant energy costs, quickly draining battery energy for example. At the outset, it is not clear which protocols minimize energy consumption, or indeed what tradeoffs between security and costs are possible. Nor is it clear whether current classes of protocols for key exchange, designed mostly with wired networks in mind, are efficient for wireless networks, or whether alternative protocols optimized for the wireless environment are needed. Quantifying these costs is essential in selecting the best candidate protocol for a wireless environment.

1.2. Our Contributions

1.2.1. Analysis Method

Our first contribution is the introduction of an analysis method to study the cost of applying security protocols over wireless channels, which are typically subject to random packet losses. This method basically transforms the protocol flow chart into a signal flow graph (SFG) to enable systematic analysis. We refer to this method as the "SFG method" throughout the text. The SFG method has the following advantages: 1) The method is easy to use since it relies on

simple widely-known techniques from Markov processes, and linear systems, yet it is general enough so that any security protocol that runs through random retransmissions (due to bad connectivity, hostile environment, etc.) can be analyzed with the SFG method. 2) The method is flexible enough to accommodate changes such as changing strategy of system nodes, fine tuning of costs, etc. by adding more nodes and branches to the SFG. 3) The method completely characterizes the distribution of the overall cost, so it can provide any statistics of interest (expected cost, variance, tail probability), and most importantly, it enables a transparent view of how the underlying security protocol affects the overall cost which provides valuable insight as described next.

1.2.2. Cost Evaluation of Key Exchange Protocols

Through the application of the SFG method, we obtain new insights on the efficiency of 'classical' cryptographic protocols for key exchange by comparing their performance in a wireless jamming environment. For example we show that, contrary to what might be expected from [5], explicit authentication of individual protocol messages via digital signatures is not the most energy-efficient approach in the face of a jamming attacker. More extreme, we show that if we are prepared to give up on forward security for our secure channel, then the very simple SOK protocol [6] is difficult to beat in terms of its communication costs. [1] Thus, from the perspective of 'classical' key exchange, our paper highlights and quantifies the security versus communication cost tradeoff under active adversaries in a wireless environment.

1.2.3. Physical Layer Augmented Key Exchange

Last, we examine the problem from a different direction: we show how state-of-the-art key exchange methods and physical layer communications techniques can be combined to thwart the jamming adversary, while maintaining communications and computational efficiency for legitimate network nodes. In particular, we augment the classical approach, and refer to our modified protocol as *physical layer augmented key exchange*. Motivated by observations obtained from our analysis, the basic idea is to move cost-expensive protocol message exchanges to an 'ephemeral' jamming-free channel, where this channel is established by repeatedly sending a short random spreading code over the open channel (instead of longer protocol messages). As opposed to the classical approach, the attacker needs to successfully eavesdrop this message to get the code for the ephemeral channel and disrupt the later exchanges. We then analyze the performance of this approach and show that this modified protocol can provide cost savings when compared with the classical protocols.

[1] A justifiably skeptical reader might question the abandonment of forward security here. However, once a secure spread-spectrum channel is established, it is easy to efficiently arrange for forward security by simply running an unauthenticated Diffie-Hellman key exchange over the channel.

1.3. Related Work

1.3.1. Modeling Using Signal Flow Graphs

A dynamic system modeled by a Markov process can be analyzed via signal flow graphs (SFGs) [7, 8]; hence, SFG-based methods have found application in the performance evaluation of numerous computer and communication systems. The work in [9] analyzes the throughput and delay of ARQ protocols using a signal flow graph tool, where the system goes from one state to the other depending on whether an ACK is received or not for a transmitted packet. The performance of 802.11 MAC layer protocols are evaluated in [10] by modeling the exponential backoff procedure with an SFG, while [11] studies the performance of ALOHA methods. What is common in these and other similar protocol analyses is that execution of the protocol passes through stages in a probabilistic manner and reaches the final stage by incurring a random cost, which is the metric of interest. To the best of our knowledge, none of the previous work using signal flow graphs was concerned with security protocols or key exchange.

1.3.2. Cost Evaluation of Security Protocols

The cost of implementing security protocols has been considered in previous literature, especially for wireless, battery-operated systems. Studies in this area include [12, 13], which show how different cryptographic methods compare in terms of the incurred communication and computation costs. These analyses demonstrate the important cost vs. security tradeoff; however, they are not based on a systematic study of the protocol flow and the probabilistic nature of wireless communication is not explicitly considered in the communication cost analysis. Formalized models of security protocols have been established using various tools [14] including dynamic system representations [15]. However, these models are most often used for evaluating the security of the protocol, i.e., for protocol verification [16]. The cost of cryptographic protocols has also been studied using formal models [17], however, again without considering the probabilistic nature of the flow taken by the security protocol.

1.3.3. Key Establishment for Wireless Channels

In this paper, we use key establishment over open channels as the problem under evaluation. For this problem, the "circular dependency" between sharing a key and avoiding jamming was recently highlighted in [5], where the authors note the fundamental need to exploit randomness to break out of this circular dependency. In particular, they propose the technique of *Uncoordinated Frequency Hopping (UFH)* where legitimate nodes send and listen on independently chosen random bands, until enough coincidences (each happening with some probability p) are reached to share all message fragments for key exchange. Follow-up work to [5] ([18], [19] among others) focuses on decreasing the communication cost of key exchange, but does not provide any systematic means to analyze the cost. In fact, the SFG method introduced here can be used to analyze the performance of these methods in a more rigorous way, as these protocols also run through a probabilistic flow of retrials.

The line of research initiated by [5] is more related to the physical layer augmented key exchange proposed here, where we make use of the 'natural randomness' provided by the wireless fading to break out of the circular dependency. Compared to UFH and related methods, our method has the advantage of not requiring any changes to the way messages are sent/received, as opposed to the mentioned methods, which require message-splitting, and the reassembly of randomly arriving fragments. However, our method assumes wireless fading for all nodes, which, in some scenarios, may be altered by a powerful attacker who can keep a strong and constant (non-fading) wireless channel to the legitimate nodes. In such a case, one should resort to creating 'artificial randomness' in some way, as done in UFH and similar methods.

1.3.4. Exploiting Wireless Fading for Security

Finally, exploiting fading for creating common randomness has been considered in previous works about the so-called "wireless wiretap channel" (see [20] and related works). The basic idea is that it is possible to deliver secret bits over the wireless fading channel in the presence of an eavesdropper "even if the eavesdropper channel is on average stronger than the main channel". We note that although we exploit fading to deliver the secret ephemeral channel code, our work is different in two ways: 1) We use fading to help reduce the cost of *cryptographic security*, as opposed to information-theoretic secrecy being the final goal in the mentioned works. 2) As the eavesdropper channel is not known, it is in general not guaranteed if indeed the delivered bits are kept secret (i.e., secrecy outage did not occur). However, in our case, the attacker is *forced* to reveal her possession of the key by jamming the ephemeral channel (and expend energy).

Packet losses over wireless channels are exploited in [21] for 'maintaining' the security of a shared key, while [22, 23] considered the use of wireless fading to 'establish' a secret key by exploiting the property that a wireless fading coefficient between two nodes is reciprocal, random and spatially independent, from which a secret key can be extracted. However, it was argued recently in [24] that extracting a secret key from fading coefficients can be highly inefficient under signal jamming. Thus, this technique is best suited only for limited scenarios involving a passive adversary and low external noise. Finally, for the implementation of our modified protocol, we borrow physical layer tools used in wireless information-theoretic secrecy literature such as cooperative jamming [25], [26].

The paper is organised as follows: Section 2 presents the system model including the attacker model, Section 3 details our analysis method (the SFG method), Section 4 introduces the key exchange algorithms evaluated in Section 5. In Section 6, we present the physical layer augmented key exchange protocol and evaluate its performance. Section 7 presents our conclusions.

Physical Environment / Wireless Channel
Environment rich in scattering, obstacles (e.g., urban)
Frequency-selective Rayleigh fading channel
Wide bandwidth available (needed for low-mobility environment)
Frequency-Hopped Spread Spectrum (FHSS) comm. system
Dynamic environment (mobile, or wide bandwidth available for static)
System nodes can transmit or receive at a given time (i.e., half-duplex)
Summary: Wireless fading channels between all nodes.

Attacker Capabilities
Unbounded energy (plugged into a wall)
Finite transmit power (could be larger than system nodes)
Cannot keep line-of-sight to a receiver for extended periods of time
May transmit and receive at the same time (i.e., full-duplex)
Separated from the system nodes by a certain minimum distance
Computationally bounded
Summary: Attacker more capable than system nodes, but cannot impose a static deterministic environment.

Table 1: System Model Assumptions

2. System Model

In this section, we provide a system model for the wireless scenario which we use to demonstrate the analysis technique, and evaluate key exchange protocols. The system model consists of a physical model (environment, and wireless communications), and an attacker model. The assumptions in our model are summarized in Table 1, and details are given below. Wireless channels nearly always exhibit random time-varying behavior. This randomness is due to a variety of sources such as variation in distance, blockages, antenna pointing, multipath fading, etc. For our analysis, we pessimistically assume only the last of these, and consider a *wireless fading channel* [27, Chapter 2], explained and modeled in detail in the following. For the attacker model, we assume the attacker may be more powerful than system nodes, but cannot impose a *non-fading environment*, e.g., by following one of the receivers and keeping constant line-of-sight, while also mitigating any reflections (e.g., using a very focused directional antenna).

This system model would be suitable for an urban environment with dynamic nodes (and/or objects around), e.g., as shown experimentally in [28], the wireless channel shows significant fading in signal measurements in similar campus environments. An example is military networks in an urban setting, which has recently gained significant research interest (channel measurement results confirming fading models for such can be found in [29], and detailed information can be found in [30, Chapters 15-17]). In contrast, this model is not applicable to an open space environment with no scatterers and with static nodes, where

the channels are constant between nodes, and a message is received with probability one without jamming, and with probability zero with jamming. Overall, our model leads to the basic assumption that a message transmitted by one node is received by the other node with some non-zero probability, therefore causing a security protocol to be completed in a succession of retrials.

2.1. Attacker Model

Consider three entities, A, B, and E, where A and B are legitimate nodes that want to communicate over an *open wireless channel* while E is an attacker that wants to eavesdrop/disrupt this process. By an "open" wireless channel, we mean a wireless channel whose parameters (e.g., frequency, power, channel encoding, etc.) are publicly known. We assume that E can try to listen to messages exchanged between A and B, and, fitting the scope of the paper, mount attacks only through the wireless channel. In particular, E can transmit her own data, fake messages and/or a random noisy signal, however, all of E's transmissions occur through the wireless channel and are subject to the random phenomenon described in detail below.

We assume that E has bounded transmission power (i.e., an energy expenditure of some finite β per unit time), as this is a basic hardware limitation of all radio transmitters. However, we assume that there is no limitation on the total energy expended by E, in other words, whereas A and B may be battery-operated wireless nodes for which limiting energy expenditure is paramount, E can be plugged into a wall outlet and thus have no concerns about battery lifetime. We assume E is full-duplex, i.e., can send and transmit signals simultaneously.

Finally, E can be located anywhere, but we assume that there is a non-zero distance between E and each of the legitimate nodes, in other words, there is a "safe range" around A and B in which E cannot be located. At first glance such an assumption seems like a weakness relative to attacker models prescribed for wired environments that employ an "attacker everywhere" approach, but it simply avoids the singularity that occurs if E is located exactly at A or B's location (see (1)), which cannot occur in practice. For many of our envisioned applications (e.g., communication between two dismounted soldiers), it is clear that an attacker extremely close to A or B can be detected and potentially eliminated by other means. Finally, it will be clear from the numerical results that the main conclusions of this paper are accentuated as the safe range becomes smaller.

As summarized above, the basic limitation we consider on the attacker capabilities is that the attacker cannot impose a non-fading environment. Hence, we assume the attacker cannot constantly follow a legitimate node and maintain line-of-sight while also mitigating all multipath. If the attacker has directional antennas, this will improve the attacker performance in a static environment, but may not help (and can even decrease jamming performance) in a mobile setting. Finally, the attacker may have multiple antennas, or there may be a number of attackers. Although this improves jamming capabilities by increasing

diversity (and total jamming power), the channel variation will still be available, but it causes a smaller success probability for system nodes. In short, the attacker assumptions are tied to the physical environment in that the attacker can be more powerful/better-equipped but cannot impose deterministic (non-fading) channels to the nodes.

2.2. Physical Model

We assume a physical environment rich with scatterers providing a number of transmission paths causing multipath fading. In this section, we describe the multipath fading phenomenon and show how it is modeled. In particular, we show how the received power over a wireless fading channel, i.e., channel strength, is modeled mathematically as a random value, and how the probability of successful reception over a wireless fading channel is calculated.

We model two main effects that impact the received power in the wireless environment: large scale path-loss caused by the distance between the transmitter and receiver, and small scale multipath fading caused by signal reflections.

First, consider *path-loss*. When a transmitter transmits, the signal spreads out in space and the power density decreases the further one is from the transmitter. Hence, the energy received without any other impairments at a node B from transmission by a node A is given by [27],

$$E_B = E_A / d_{AB}^{\alpha} \tag{1}$$

where E_A is A's transmit energy, d_{AB} is the distance between node A and node B, and α is the "path-loss exponent", which generally ranges from 2 to 4 depending on the environment.

However, the primary impairment impacting the signal in a wireless communication system is *multipath fading*, which is caused by the reflections of the transmitted signal off objects in the environment. The differences in the lengths of the paths followed by these reflections relative to the direct signal and, hence, their relative phases, cause the reflections and the main signal to add destructively or constructively at various points in space. A receiver essentially takes a sample of this spatial pattern at its location, and its received power is a value that can be above or below the average that would be expected at that location based on the path-loss, given by (1). The random variation in received power is also time-dependent. This is due to the signal taking different paths in time caused by movement of the receiver and/or other objects in the environment. Hence, the actual received signal energy is a random value that depends on location, time, and (as described next in detail) the frequency of the transmitted signal.

As will be clear, we are interested in two basic wireless communication methods: *narrowband* and *wideband* communications. Multipath fading affects narrowband and wideband signals differently. A narrowband signal is one that occupies only a narrow frequency range, an example of which is the early analog cell phones of bandwidth 30 kHz. The spatial fading pattern described above

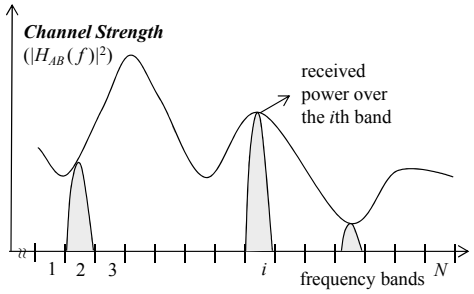


Figure 1: Illustration of wireless fading as a function of the frequency of the transmitted signal. A wide bandwidth system is shown with N (narrow) frequency bands. The received power of a narrowband signal transmitted over the ith band is equal to the transmitted power times the fading value of the ith band, and the fading value for each frequency band is independent. In contrast, for wideband communication, a single message is sent using many bands, essentially averaging out the fading values of the bands used. Also note that, in a mobile setting, the whole plot above changes from one time slot to the next.

also depends on the frequency of operation, as the fading patterns at two frequencies that are close in value is correlated. For example, in a typical urban outdoor environment, a 100 kHz separation between two frequencies is required before the spatial multipath fading patterns at these two frequencies can be assumed to be independent [31]. Hence, a narrowband signal will be impacted by essentially a single fading value corresponding to the frequency at which it operates around (see Fig. 1).

A wideband signal is one that occupies a much larger frequency range, an example of which would be more recent spread spectrum cell phones with bandwidths on the order of 1 MHz. In contrast to narrowband signals, wideband signals, if designed properly, can achieve performance for an equivalent channel that is the average of a large collection of fading values. By the law of large numbers, each instantiation then approaches its expectation, and the fading is essentially eliminated by averaging.

In this work, we assume that a "wide bandwidth system", e.g., a Frequency-Hopped Spread Spectrum (FHSS) system [32], is available, where narrowband and wideband communications are both possible. In other words, one can choose to send a message over a single frequency band, or send a single message using ("hopping between") many bands in a specified order given by a "channel code". We describe next how the probability of successful reception is calculated for both narrowband and wideband signals.

For mathematical analysis, we consider the Rayleigh fading model [31], which has been used extensively in the analysis of wireless systems and is applicable in environments without line-of-sight paths and with a large number of signal reflections [27]. We assume a block fading model, where each message transmission happens over a time duration smaller than the coherence time of the channel (that is, the interval over which the channel fading gain remains relatively constant), and fading is independent from one message transmission to

the next. Independent fading can be achieved by spacing transmissions in time larger than the coherence time of the channel (e.g., for urban environments this would be on the order of 10's of msec), or by selecting a different narrowband from the spectrum (see Fig. 1). It is important to note that, while we use Rayleigh fading for performance analysis in the paper, our methodology is not limited to this model and is applicable in general to any wireless fading channel, albeit under a new set of probability calculations for message reception.

Modern communication systems demonstrate a threshold effect in packet-data communication based on the received signal-to-interference-plus-noise ratio (SINR): a packet is received with high probability if the SINR is above a given threshold γ, and the packet is lost if the received SINR is below γ.

For narrowband communication, in the Rayleigh fading model, the random power variation caused by multipath fading is modeled as an exponential random variable. (Detailed explanation is omitted due to space constraints, more information can be found in, e.g., [32, Chapter 14].) Let $P_{\text{rcv}}^{(B\leftarrow A)}$ be the probability that a packet sent from A is received at B; then, it can be shown that

$$P_{\text{rcv}}^{(B\leftarrow A)} \quad = \quad \exp\left(-\gamma \frac{N_0}{E_A} d_{AB}^{\alpha}\right) \tag{2}$$

for a transmission from A to B on a narrowband channel, where N_0 is a parameter proportional to the power of the thermal noise in the receiver.

In this work, we are also interested in this probability when the attacker also sends a jamming signal over the narrowband channel. The jamming power received from E is also subject to fading and hence modeled as an exponential random variable, but with a different parameter, giving the successful reception under jamming as

$$P_{\text{rcv}}^{(B\leftarrow A)}|\text{Jamming} \quad = \quad \frac{\exp\left(-\gamma \frac{N_0}{E_A} d_{AB}^{\alpha}\right)}{1+\gamma \frac{E_A/d_{AB}^{\alpha}}{\beta/d_{BE}^{\alpha}}} \tag{3}$$

Notice that, as expected, reception is more likely to succeed when the attacker's transmit energy β is lower or her distance to B, d_{BE}, is larger.

The probability values for wideband communication are found next. Per above, a well-designed system for a wideband channel will mitigate the multipath fading through averaging, hence, the packet is either received or not based on the path-loss incurred on the transmission. Hence, for a wideband wireless channel:

$$P_{\text{rcv}}^{(B\leftarrow A)} = \begin{cases} 1, & \text{if } \dfrac{E_A/d_{AB}^{\alpha}}{N_0} \geq \gamma, \\ 0, & \text{if } \dfrac{E_A/d_{AB}^{\alpha}}{N_0} < \gamma. \end{cases} \tag{4}$$

In the case of jamming, probability values depend on whether the attacker knows the code to the wideband channel. If the code is secret, the above probabilities

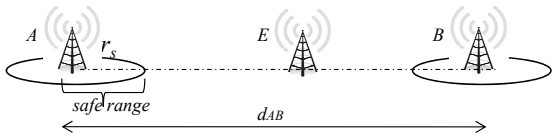


Figure 2: Abstracted view of the network, for the pessimistic case where A, B and E are located on a line. A, B are the legitimate nodes with a distance d_{AB} apart. Both have some certain safe range with a radius r_s surrounding them. E can be located anywhere outside safe ranges, however it is optimal for E to be located on the line to deliver maximum jamming power, i.e., $d_{AE} = d_{AB} - d_{BE}$, so $d_{AE} \in [r_s, d_{AB} - r_s]$. Note that the network is shown with only the nodes and not the physical environment, which includes scatterers, obstacles, etc. which leads to a wireless fading channel as described in detail in Section 2.2.

still apply. However, if the code is known, attacker jams the communication with probability one, as fading is averaged out also for the attacker.

In summary, we assume the success probability under jamming is *i)* a non-zero value given by (3) for a known narrowband channel, *ii)* zero for a known wideband channel, *iii)* one for a private wideband channel.

2.3. Network Model for Analysis

For the cost evaluations done in Sections 5, 6, we assume a network where A and B are at fixed locations a distance d_{AB} apart. Per above, A and B are each contained in a disk where E is not located. The radius of this disk, r_s, is an important parameter, which is directly related to the amount of jamming from E that is incurred at A or B. E can be anywhere outside the disk. In the homogeneous propagation environment that we assume in this paper, where the large-scale path-loss of the signal is only a function of distance, E's ability to eavesdrop and jam transmissions in the wireless environment is maximized when she lies on the line between A and B, hence we assume a linear network model with all three nodes on a single line as given in Fig. 2.

Note that this network model is simple enough for probability and cost calculations needed for the analysis, but it also captures the wireless fading parameters and assumptions in the physical model. For example, the attacker's given location on the line gives the distances between all pairs, while varying the attacker location provides the effect of varying jamming power suffered at the nodes. Other considerations such as attacker's transmit power, or multiple attackers can be reflected in the model by increasing the amount of jamming power in calculations accordingly, or by adding diversity coefficients, etc. Also, for keeping calculations simple, nodes are assumed static in the network model, but note that the independent fading values assumption in the model is still valid due to the use of a different frequency band each time A sends her message.

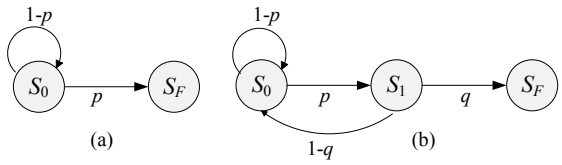


Figure 3: Two process are shown with their Markov state diagrams. State transitions occur with the displayed probability, and each take unit time. (a) A process with initial state S_0, and final state S_F. Completion time X_1 is a random variable with geometric distribution with parameter p. (b) A process with initial state S_0, intermediate state S_1, and final state S_F. Compared to X_1, it is harder to find the distribution for completion time X_2 by inspection.

3. The SFG Method

3.1. Introduction

The execution of a cryptographic protocol over a wireless channel can be modeled as a process that probabilistically passes through a number of states (e.g., see Fig. 9). From any given state, the next step depends on the occurrence of some random event based on the wireless channel. The participants A and B incur a cost at each step, and the overall cost of one execution of the cryptographic protocol is a random value, which is the value of interest. Often, more statistics than the simple 'average cost' are of interest; for example, a system designer may be interested in how likely it is for the amount of battery energy spent executing a given security protocol to remain within a specified range.

As a motivating example, consider the two Markov state diagrams shown in Fig. 3 representing two simple processes, where state transitions take unit time. The first process has only an initial state S_0 and a final state S_F. The completion time of this process X_1 is geometrically distributed with parameter p, the probability of moving from S_0 to S_F. Hence, any statistics of interest can be found easily from the distribution. For the second process (Fig. 3 (b)), there is an additional intermediate step S_1. Note that from S_1, the process can move to S_F or come back to S_0 to restart the process. The completion time X_2 for this process is harder to characterize by inspection, although some statistics, e.g., $E(X_2)$ can be found without obtaining the whole distribution. Hence, the conclusion is that even a process with a small number of states, but with *feedback loops* may be hard to analyze, and one needs a systematic way of evaluating an arbitrary process.

Since the cryptographic protocol running over a wireless channel is a dynamic probabilistic system, it can be modeled by a Markov process. The states in the protocol (e.g., boxes in Fig. 9) correspond to states in the Markov process. The branches connecting the states of the protocol correspond to the state transitions of this Markov process, and each state transition has an associated probability, which, in our case, is determined by the wireless communications model. Also, each transition potentially incurs a cost (e.g. energy and/or delay) to A and B. Our aim is to study the Markov process and find the distribution of its

overall cost from the initial state to final state. To accomplish this, we map the Markov state diagram to a signal flow graph [8], where states become nodes and state transitions become linear systems that act upon the signal, as described in detail in the next section. Once the Markov process is mapped to a signal flow graph, reduction methods can be used to find the transfer function between the input and output signals, as described in detail below. This transfer function is the moment-generating function (mgf) of the cost, and it will have high utility in answering questions of interest.

Definition 1. *Let X be a real-valued random variable. The moment-generating function of X is defined as*

$$M_X(s) = E(e^{sX}), \quad s \in \mathbb{R} \tag{5}$$

where $E(\cdot)$ corresponds to expectation.

We are sometimes interested in the joint behavior of two random costs X, Y (e.g., energy and delay). The *moment-generating function* of X, Y is defined as

$$M_{XY}(s,t) = \mathrm{E}(e^{sX}e^{tY}) \quad s,t \in \mathbb{R}. \tag{6}$$

Note that the marginal mgf can be easily found by the following relation: $M_X(s) = M_{X,Y}(s,0)$

Being a Laplace transform, the mgf of a random variable contains all the information about its distribution and the exact probability density function can be found by an inverse transform. However, often a statistic of the cost (e.g., average energy expenditure or variance of the delay) is of interest. Then, the moment-generating function readily supplies such: for integers m and n,

$$\mathrm{E}(X^m Y^n) = \frac{\partial M_{XY}^{m+n}(s,t)}{\partial s^m \partial t^n}\Big|_{s=0,t=0}, \tag{7}$$

and the expected value of X (of Y) is just the special case with $m = 1$ and $n = 0$ ($m = 0$ and $n = 1$).

It is often of interest to know how likely it is that the cost will exceed a certain threshold; for example, a radio might have some initialization period during which the cryptographic protocol must be executed, and hence it is of interest to know the probability that the protocol is not executed within that period. We can find this by calculating the so-called "tail probability," which is readily bounded using the mgf and the Chernoff bound [33].

Chernoff Bound

Let X be a random variable, and $M_X(s)$ its moment-generating function. Then

$$P(X \geq c) \leq \min_{s \geq 0}(e^{-sc}M_X(s)), \forall c \in \mathbb{R} \tag{8}$$

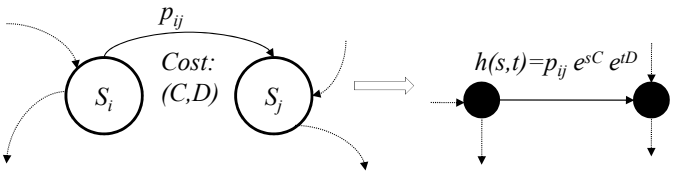


Figure 4: The Markov state diagram is mapped to a signal flow graph. Each state is mapped to a node, each state-transition is mapped to a branch between nodes.

Markov Process	*Linear System*
State Diagram	Signal Flow Graph
State	Node
Initial State	Input Node
Final State	Output Node
State transition from S_i to S_j	Signal passing through a linear system
Branch with prob. p_{ij}, cost C, D units	Linear system with transfer function $h(s,t) = p_{ij}e^{sC}e^{tD}$
End-to-end cost distribution	Overall impulse response
Moment-generating function of cost distribution	Transfer function of the whole system

Table 2: Mapping a Markov process to a linear system

3.2. Methodology

First, we map a protocol flow chart to a Markov state diagram. For example, Fig. 10 (a) provides the Markov state diagram for the protocol flow chart of Fig. 9. Next, consider an arbitrary section of a Markov state diagram as given in Fig. 4. To map this state diagram to a signal flow graph, each state is replaced with a small black circle to represent a node. The transition from state i to state j is labeled with p_{ij}, the probability of transitioning to state j from state i, and costs C, D, where C is the cost of that transition in the first metric of interest, and D is the cost of that transition in the second metric of interest. This state transition is mapped to a branch in the signal flow graph and labeled with $h(s,t) = p_{ij}e^{sC}e^{tD}$, the transfer function of the linear system represented by this branch. This mapping is summarized in Table 2. In the protocol evaluations, the transition probabilities will be found from the wireless system calculations of Section 2.2, whereas the costs will be clear from the protocol description.

Next, the resulting signal flow graph is systematically reduced through a number of transformations as shown in Figure 5 until a single branch from the initial state to the final state is obtained. An example of such a reduction is shown in Figure 10. The label of the single resulting branch corresponds to the joint mgf of X and Y, from which moments of the cost of execution of the protocol are readily obtained from (7).

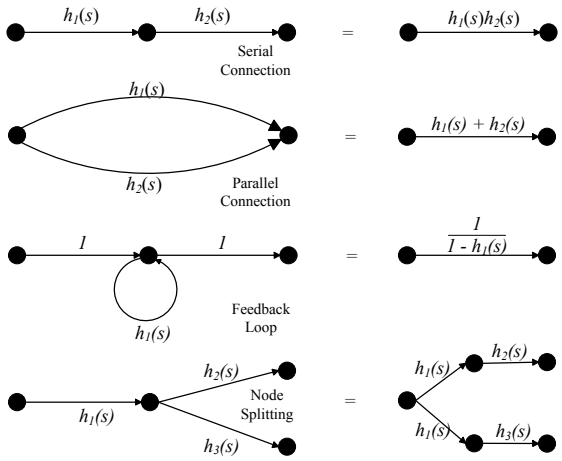


Figure 5: Basic equivalences in signal flow graphs.

4. Key Exchange Protocols

A key exchange protocol provides a mechanism by which two parties A and B can generate a common secret key (or session key) while communicating over an insecure channel. Many different security models and security definitions for key exchange protocols have been developed by the cryptographic research community (see for example [34, 35, 36, 37]). A consensus has now emerged around a few essential security properties. *Session-key security* refers to the property that the compromise of one (or many) session key(s) should not affect the security of other session keys. *Forward security* refers to the property that past session keys are not compromised even if the long-term keys of the parties are. The prevention of *unknown key-share attacks* and resilience to *key-compromise impersonation attacks* are also considered important goals. For our analysis, an important characteristic will be whether the messages in the protocol are *explicitly* authenticated or not. In the former case, each message is protected by a digital signature and a time-stamp. In the latter case, the parties obtain an *implicit* key authentication property: each party is assured that only the other *could* generate the same session key.

We will consider protocols in both the traditional PKI setting and in the identity-based setting (where each node is associated with an identifier from which its public key can be derived, while a trusted authority (TA) pre-provisions each node with the private key corresponding to its identifier). To maintain a level playing field for fair comparison of the different protocol types, we will assume that all nodes have their public keys certified by a single CA in the PKI setting, and that this CA's public key is hard-coded into the network nodes. We will also assume that certificates consist solely of the CA's signature on the nodes' public keys. (In reality, certificates are much larger and more complicated data structures than this.) Similarly, we will assume, in the identity-based

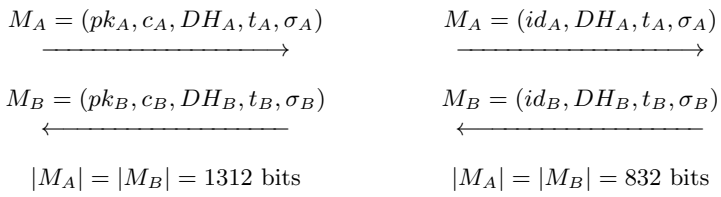


Figure 6: Protocol 1: PKI-based Diffie-Hellman (left); Protocol 2: Identity-based Diffie-Hellman (right)

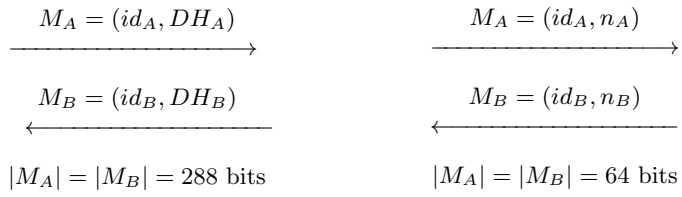


Figure 7: Protocol 3: Forward secure, implicitly authenticated, identity-based (left); Protocol 4: SOK with nonces (right)

setting, that each node obtains its private key from a single TA and that this TA's public parameters are hard-coded into the network nodes. In addition, we assume that, since the networks we study are large and dynamic, each node holds its public key (or identifier) and corresponding private key, but does not know *a priori* the public keys (or identifiers) of other network nodes. In the identity-based setting, a node is authenticated by effectively proving knowledge of the private key matching its claimed identifier during the key exchange protocol. Through the pre-provisioning of private keys, the TA defines the set of legitimate node identifiers. In this way, legitimate identifiers are distinguished from bogus ones potentially injected by the attacker. Similarly, in the traditional PKI setting, it is the set of nodes that can present a valid certificate and prove knowledge of a private key matching the public key in the certificate that are considered legitimate.

We take 128 bits as the target security level for all our key exchange protocols. This is appropriate for the protection of, for example, classified information in a tactical military network. We make extensive use of elliptic curve and pairing-based cryptographic techniques, including sufficient references to enable the interested reader to verify our parameter choices.

We start by considering a classical PKI-based Diffie-Hellman key-exchange protocol (Fig. 6, left). This protocol is *forward secure* and the protocol messages are *explicitly authenticated*. In this protocol, A sends B a message including pk_A (A's public key, 256 bits), c_A (a certificate on her public key, 256 bits), DH_A (a Diffie-Hellman value, 256 bits), t_A (a time-stamp, 32 bits) and σ_A (a signature

on the whole message, 512 bits). Here we use a BLS signature [38] on a BN curve [39] for the certificate. This allows us to minimise the size of the signature in the certificate at 256 bits and exploits the fact that we do not need to transmit the CA's public key. We use the ECDSA scheme for the signature σ_A, in order to minimise the sum of the size of this signature and the corresponding public key whilst avoiding the relatively expensive pairing calculations that are needed in the BLS scheme. The Diffie-Hellman exchange takes place over a fixed elliptic curve group whose elements can be represented by 256 bits, using point compression. Similarly, B sends message M_B to A of the same form; A and B can then create their session key by applying a key derivation function to the shared Diffie-Hellman value and their public keys. The total cost per message is 1312 bits. We note the requirement of this protocol that the protocol participants have synchronized clocks or access to a coordinated time service.

We next consider an analogous protocol in the identity-based setting. We consider a *forward secure* identity-based Diffie-Hellman key-exchange (see Fig. 6, right), whose messages are *explicitly authenticated*. Here, A sends B a message including her identifier id_A (32 bits), a Diffie-Hellman value DH_A (256 bits), a time-stamp t_A (32 bits) and an identity-based signature σ_A on the preceding fields (512 bits using the BLMQ scheme [40] over BN curves). Here, an exchange of short identifiers replaces the exchange of public keys and certificates from the previous protocol. The exchanged messages are each 832 bits long. Again, this protocol requires access to a coordinated time service. We note that, through the use of time-stamps, these first two protocols share the same synchronization and anti-replay properties.

Our third protocol, the SCK-2 protocol from [41], is an *implicitly authenticated*, *forward secure* identity-based protocol. Here A sends B a message of the form $M_A = (id_A, DH_A)$ and B sends a similar message M_B to A (see Fig. 7, left). The session key is calculated by combining the Diffie-Hellman private and public values, the identities and the private keys in a particular way that is detailed in [41]. A number of other protocol designs are available, with [42] providing a useful survey of the available alternatives. We have selected the SCK-2 protocol because of its low bandwidth consumption (here, the protocol messages are only 288 bits each at the 128-bit security level) and its proven security properties [41, 42].

Our fourth protocol, the SOK protocol [6], is also *implicitly authenticated*, but sacrifices the forward security property in order to reduce bandwidth to a minimum. In the basic version of this protocol A simply sends her identifier id_A to B and B sends his identifier id_B to A; the two parties then combine their respective identifiers and private keys in a specific way to obtain a shared session key. We augment this basic protocol with 32-bit nonces (see Fig. 7, right), with these nonces being included in the key derivation step, in order to prevent the agreed key from being a static value. The exchanged messages are still very short, just 64 bits each. To instantiate this protocol efficiently at the 128-bit security level, we use BN curves and asymmetric pairings, equipping each party with two private key components, one in each group input to the pairing operation and using an ordering on node identifiers to determine in

	Setting	Message size (bits)	Authentication	Forward security
1	PKI	1312	Explicit	Yes
2	ID-based	832	Explicit	Yes
3	ID-based	288	Implicit	Yes
4	ID-based	64	Implicit	No

Table 3: Comparison of key exchange protocols.

which order hashed identifiers/private key components are input to the pairing operation. These modifications do not affect the bandwidth consumption of the protocol. Basic versions of this protocol (without nonces) were proven secure in [43, 44].

Protocols 3 and 4 have the property that an attacker can easily spoof the first (or second) protocol message, forcing a legitimate party to expend energy on communication and computation. However, the attacker cannot then go on to compute the appropriate session key, so this kind of attack will eventually be detected when the legitimate receiver of a spoofed message computes the session key and switches to the secure channel determined by that session key. Yet, this property would appear to leave Protocols 3 and 4 vulnerable to resource exhaustion attacks. This is indeed the case, but this is fully accounted for in the specific signal flow graph we use for analyzing Protocols 3 and 4. For example, from the flow chart in Fig. 11 we can see that a spoofed message will lead to A and B not meeting on a secure channel, causing extra communication, a certain amount of time to elapse, and the protocol to then be re-run. We also note that even explicitly authenticated protocols like Protocols 1 and 2 can be vulnerable to resource exhaustion attacks. For example, the attacker can easily force a receiver to perform a signature verification operation simply by sending a message of the correct format (but with a random value for the signature field). Such an attack would be detected as soon as the signature is checked and so the receiver would not go on to send a response message in that protocol run – hence, the protocol run would be aborted early, in contrast to the situation with Protocols 3 and 4. This advantage of explicitly authenticated protocols is again reflected in our signal flow diagram for Protocols 1 and 2 (see Fig. 9). However, we note that this advantage is not cost-free, since it requires the transmission of digital signatures along with protocol messages.

In Table 3 we summarize the properties of the protocols considered so far.

5. Evaluation of Key Exchange Protocols

In this section, we study the performance of classical key exchange protocols in a wireless environment. The description of the protocol steps is given in Fig. 8. The message transfers for the key exchange protocol are done over a public narrowband channel, referred to as open air channel. Once a key, K_{AB}, is generated, A and B use it as a code to establish the secret wideband channel $Ch(K_{AB})$ over which future communications will take place.

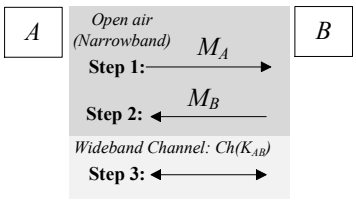


Figure 8: Description of the key exchange protocol over a wireless channel. A and B exchange messages M_A, M_B over the publicly known narrowband channel referred to as open air. Then they switch to a wideband (spread spectrum) channel given by the session key K_{AB}, where they can communicate efficiently and free from jamming attacks as long as K_{AB} is secret. Protocol messages are described in Figs. 6, 7. Communication over the narrowband and the wideband channel is modeled in Section 2.2.

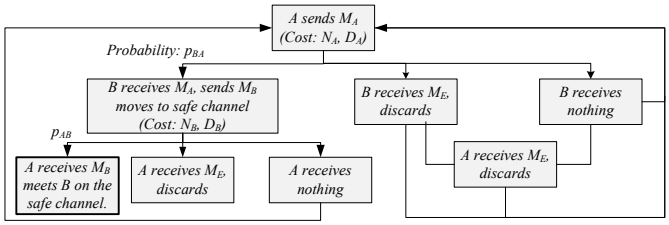


Figure 9: The flow chart for Protocols 1 and 2.

5.1. Cost Analysis

We divide the protocols in Table 3 into two classes: (1) protocols with explicitly-authenticated messages, and (2) protocols with implicitly-authenticated messages, since the flow charts associated with protocols in each class are identical except for different costs on the branches.

5.1.1. Protocols with explicitly-authenticated messages

The flow chart common to both Protocols 1 and 2 is given in Fig. 9. Since each message is explicitly authenticated, there is no advantage to E in inserting a fake message; thus, her optimal strategy is to transmit random jamming noise to minimize the probability of message reception. Hence, when E hears the transmission of messages M_A and M_B, she transmits random jamming noise. A and B accept only the messages that they correctly receive and can be authenticated, otherwise the protocol re-starts.

Following the SFG method, we first convert the flow chart into a signal flow graph and then simplify it to obtain the transfer function. A simplified Markov state diagram and signal flow graph for Protocols 1 and 2 is given in Fig. 10. The functions $h_1(s,t), \cdots, h_4(s,t)$ denote the transfer functions for each branch. Probabilities p_{BA} and p_{AB} denote the respective probabilities of B successfully receiving M_A and A successfully receiving M_B (found using the communication

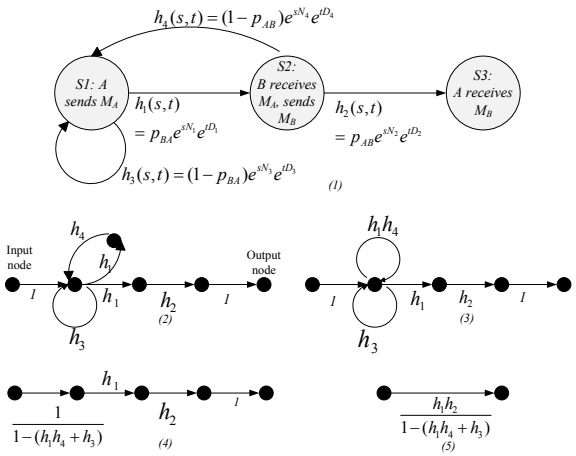


Figure 10: Markov state diagram for Protocols 1 and 2, and the corresponding signal flow graph reduced to a single branch step by step.

model in Section 2), while $(N_1, D_1), \cdots, (N_4, D_4)$ denote the respective energy and delay costs on each branch.

To calculate $(N_1, D_1), \cdots, (N_4, D_4)$ we proceed as follows. Let N_A and N_B be the energy spent transmitting M_A and M_B, respectively. Then, in the signal flow graph we have, $N_1 = N_3 = N_A$ and $N_2 = N_4 = N_B$. Likewise, if D_A and D_B are the times it takes to send messages M_A and M_B, respectively, the delay costs are $D_1 = D_A$, and $D_2 = D_B$. The branch h_3 corresponds to the case where A sends M_A, but B misses and keeps silent. If E remains silent, A will quickly sense that there is no message from B and switch to transmit mode and send M_A again, hence implying $D_3 = D_A$, and, likewise, $D_4 = D_B$. On the other hand, if E focuses on maximizing the delay in completing key exchange, she will always choose to send a random message to A even when B misses M_A to cause A to waste time. This will make $D_3 = (D_A + D_B)$.

Let X, Y be the random variables denoting the total cost in transmit energy and delay, respectively. The moment-generating function of the cost is then:

$$M_{XY}(s,t) = \frac{h_1 h_2}{1-(h_1 h_4 + h_3)}, \quad \text{where}$$

$$\begin{aligned} h_1(s,t) &= p_{BA}e^{(sN_A+tD_A)}, \\ h_2(s,t) &= p_{AB}e^{(sN_B+tD_B)}, \\ h_3(s,t) &= (1-p_{BA})e^{(sN_A+tD_A)}, \\ h_4(s,t) &= (1-p_{AB})e^{(sN_B+tD_B)}. \end{aligned} \tag{9}$$

The reduction steps in Fig. 10 illustrate how the end-to-end transfer function given in (9) is obtained by employing the equivalences in Fig. 5.

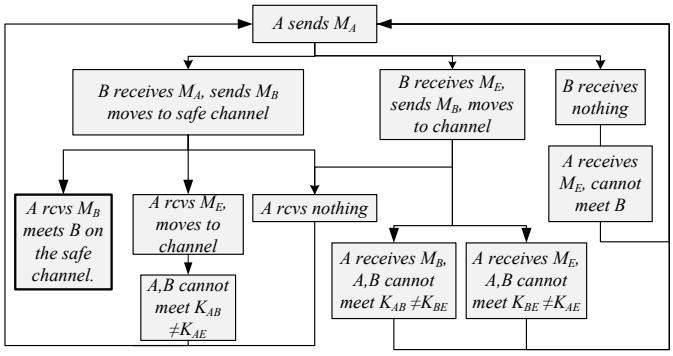


Figure 11: The flow chart for Protocols 3 and 4.

5.1.2. Protocols with implicitly-authenticated messages

As compared to protocols with explicit authentication, a differentiating feature of protocols employing implicit authentication is that they do not allow a node to immediately discard a fake message successfully inserted by an attacker. Thus, in this case, E may gain an advantage by transmitting fake messages over the open air. The flow chart for Protocols 3 and 4 is given in Fig. 11. When B receives a message M_E from E, he will reply with his message M_B and will attempt to compute a key K_{BE}; however, since E does not possess the correct keying materials she will not be able to compute the same key and the process will fail, with B eventually returning to the open air channel to listen. While not compromising security, this will incur a transmit energy cost for B, which was not present in explicitly-authenticated protocols. While B is replying to E's message, E may also send a fake message to A, which also causes her to generate some key K_{AE} and move to the corresponding spread spectrum channel; again, the process fails and A returns to the start of the protocol. While protocol restart is more likely in implicitly authenticated protocols, the advantage is that messages are shorter and the cost incurred in each cycle is lower. Thus, we can see a tradeoff in cost incurred by a protocol versus its other properties.

The signal flow graph for this case is given in Fig. 12. The additional state S_4 as compared to Fig. 10 is necessary for the case that B receives E's fake message (but, cannot be discarded yet as the message is not explicitly authenticated). Let p_{BA}, p_{BE}, and p_{AB} be the probabilities of successful message transmission between the respective nodes. Probabilities p_{BA}, p_{BE} are calculated using the insight that when A transmits M_A, E will insert a fake message instead of random jamming noise, while p_{AB}, remains the same as in the explicitly-authenticated case.

Borrowing from Fig. 10, the cost only needs to be calculated for the branches connecting S_1 and S_4. The branches labeled g_1, g_2 have energy cost $N_{g_1} = N_A$ and $N_{g_2} = N_B$. To calculate the delay costs, we assume that when B receives M_E, he will reply, so A will incur a delay receiving this message. Hence, the branch g_2 has a delay of $D_{g2} = D_B$, and, likewise, the delay for g_1 is $D_{g1} =$

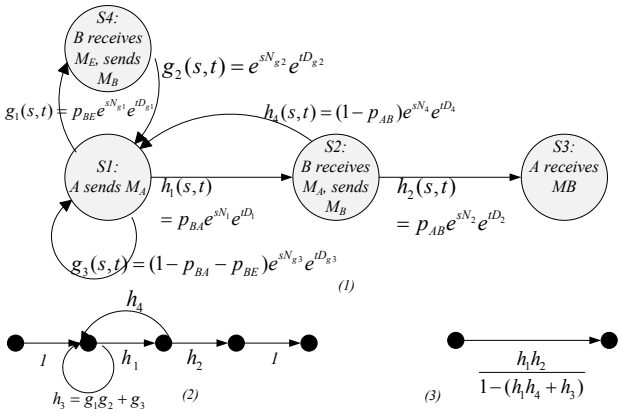


Figure 12: Markov state diagram for Protocols 3 and 4, and the corresponding signal flow graph reduced to a single branch.

D_A. The self-loop branch g_3 is identical to that in the explicitly-authenticated protocol.

The moment-generating function for the end-to-end cost is given as $M_{XY}(s,t) = \frac{h_1 h_2}{1-(h_1 h_4 + h_3)}$, where:

$$\begin{aligned} h_1(s,t) &= p_{BA} e^{(sN_A + tD_A)}, \\ h_2(s,t) &= p_{AB} e^{(sN_B + tD_B)}, \end{aligned}$$

$$\begin{aligned} h_3(s,t) &= (1-p_{BA}-p_{BE}) e^{(sN_A + tD_A)} \\ &\quad + p_{BE} e^{\left(s(N_A+N_B)+t(D_A+D_B)\right)}, \\ h_4(s,t) &= (1-p_{AB}) e^{(sN_B + tD_B)}. \end{aligned} \tag{10}$$

5.2. Numerical Results

For numerical calculations, we assume that nodes send messages with a transmit energy of 10mW (i.e., $E_A = E_B = \beta = 10$dBm), which is equal to 10mJ/s, and the symbols are transmitted at a rate of one Msymbols/sec. Hence sending each physical layer symbol costs $10^{-2}\mu$J, and takes 1μs. Other parameters for the wireless communication model are taken as follows: $\alpha = 2$ and $E_A/N_0 = E_B/N_0 = 30$dB.

5.2.1. Transmit Energy Cost

Plots are given in Fig. 13 for the expected energy expenditure of Protocols 2 and 3. For these plots, we assume a safe radius $r_s = 1$, and plot the cost as a function of E's location, d_{AE}, which varies from 1 to 9 as shown in the network model in Fig. 2. For each value of d_{AE}, we calculate the expected value of the cost using the mgfs given in (9) and (10).

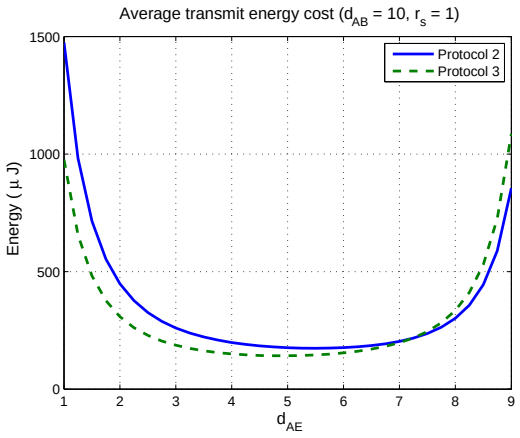


Figure 13: Cost of key establishment using Protocols 2 and 3. The cost metric is the total transmit energy spent by A and B during the protocol.

As can be seen, although they provide the same security, Protocols 2 and 3 are quite different both in their cost analysis and cost values. For example, consider $d_{AE} = 1$; using Protocol 3 results in a transmit energy saving of 33%, i.e., of more than 1.5dB, which is a significant amount by wireless-communication engineering standards. Furthermore, the comparison of Protocols 2 and 3 also illustrates an important point that simply using message sizes as the metric to judge efficiency gains can be misleading. For example, Protocol 3 has a message size of 288 bits while Protocol 2 has a message size of 832 bits; however, because of the extra cost due to the lack of explicit authentication, the overall cost reduction with Protocol 3 compared to Protocol 2 is not necessarily 65% (as implied by the ratio between the message sizes). In fact, for $d_{AE} = 9$, Protocol 2 requires *less* transmit energy on average, by roughly 20%.

Aside from comparing the protocols, plots in Fig. 13 illustrate the fact that the attacker's location is a very important parameter affecting the total cost of key exchange over wireless channels. The cost grows exponentially when the attacker is very close to either of the nodes, which is natural since the success rate of an adversary's attacks becomes higher. Since the exact location of the attacker is generally unknown to the nodes, the maximum cost over all possible attacker locations outside the safe range, i.e. the worst-case cost, may be more important. For example, for Protocol 2 in Fig. 13 the value for $d_{AE} = 1$ represents the worst-case cost.

Figs. 14, 15 compare the worst-case transmit energy costs of key exchange for varying safe radius sizes. As expected, an increased safe range results in decreased cost. Fig. 14 compares Protocols 2 and 3, and shows that there is a benefit, in terms of reducing the worst case average energy, to use a protocol with implicitly-authenticated messages in place of one with explicitly-authenticated messages. Fig. 15 (a), (b) compare the costs for Protocols 1 and 2, and Protocols 3 and 4, respectively. Fig. 15 (a) shows the benefit of switching from PKI-based

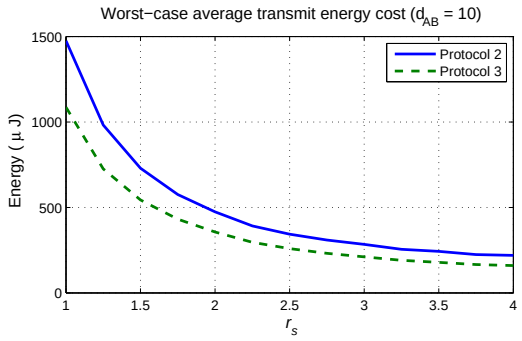


Figure 14: Worst-case cost values plotted for Protocols 2 and 3. Cost metric is total transmit energy spent by nodes. For a given safe range size r_s, E can be located anywhere outside these ranges. Hence, the place where she causes maximum cost becomes the worst-case cost for that range. For each value of r_s, the maximum value of the cost as a function of E's location is calculated and plotted.

to ID-based security keeping everything else (security level, protocol type) the same. Figure 15 (b) shows that a major gain in cost reduction is obtained by switching to the SOK protocol, which sacrifices forward security.

To summarize, *the overall combination of the changes from PKI-based to ID-based, then explicit to implicit, then sacrificing forward security, shows almost a 10-fold improvement in energy consumption.* This new insight will be very helpful for system designers in fairly judging the security versus cost benefits of different protocols. Furthermore, one can concretely see that the intuitive approach of achieving small message sizes is crucial to building efficient key exchange protocols for wireless channels.

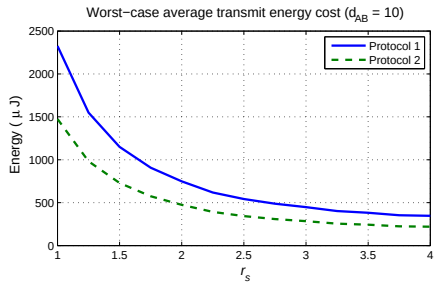
Finally, the above plots can be reproduced for different values of E's transmit power, which have been omitted due to space limitations. As expected, the cost increases as a function of the attacker's transmit power; however, the trend as a function of the attacker's location is unchanged.

5.2.2. Delay

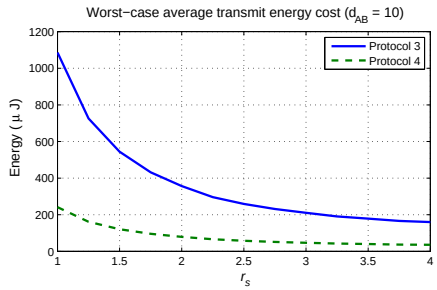
For delay, a statistic of greater interest than average delay is how likely it is for the key exchange to complete within a certain time duration. In other words, we are interested in the probability that the delay exceeds a certain threshold. For numerical results, we use the mgfs given in (9) and (10) to find the bounds on the tail probability using the Chernoff bound given in (8). Fig. 16 plots the bounds for Protocols 2 and 3.

5.3. Extensions

In the above analysis, we studied key exchange protocols with a simplified model, enough to highlight crucial trade-offs involved in the selection of a wireless key exchange protocol. We note, however, that the SFG method is flexible enough to accommodate more realistic considerations. For example, the cost



(a)



(b)

Figure 15: Worst-case cost values for Protocols 1 and 2 in (a), Protocols 3 and 4 in (b). Calculation is the same as the case in Fig. 14.

of receiving a message, switching between operating frequencies, etc. was neglected in the above analysis. These cost values can be added with obvious changes to the protocol flow chart and the corresponding SFG. Moreover, the cost on each branch can be modeled as a random variable (e.g., backoff time), rather than some fixed deterministic value as assumed above. Then, the transfer function on that branch will be the mgf of the cost random variable, with no change to the analysis. Other changes in the model may be handled by adding more states to the flow chart, i.e., more nodes, branches to the SFG. Although this may complicate the expression for the overall transfer function, the basic reduction steps all remain the same. For example, for practical scenarios, the existence of an attacker itself may be modeled as a probabilistic event, which causes the process to take different paths depending on the states "attacker exists" or "no attacker". In the same manner, adaptive strategies for nodes can be incorporated (e.g., switch to a higher transmit power level after a packet loss), or for an adversary with limited resources, the SFG method can be used to find the adversary's optimal attack.

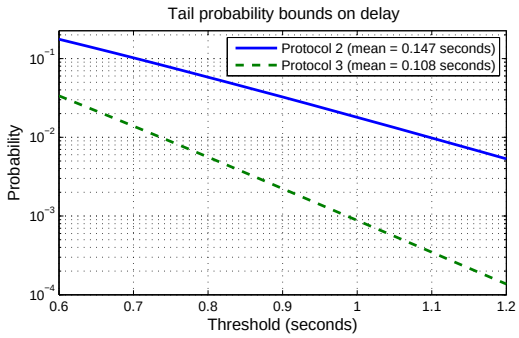


Figure 16: Bounds on tail probabilities for Protocols 2 and 3 with E keeping silent if she senses no message M_B. For each r_s, we calculate the bound for the worst-case point of the expected value. Although, the bounds calculated for the two protocols differ, we hasten to note that making a comparison of protocols based on bounds can be misleading and the curves should be considered individually.

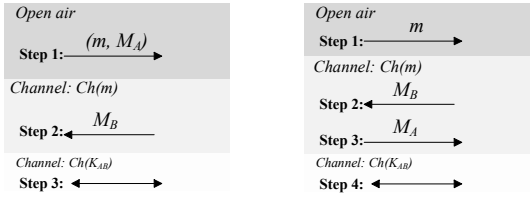


Figure 17: Modified protocol (left): Sending a random ephemeral spreading code appended to the first message. Modified protocol (right): Sending a random ephemeral spreading code over the open air. Note that, for Step 1, at each trial a different narrowband channel is used, hence nodes essentially "probe" the whole bandwidth for a good pair of fades (see Fig. 1).

6. Physical Layer Augmented Key Exchange

6.1. Protocol Modifications

One observation from the previous analysis is that the large costs are incurred because all of the messages need to be carried over the open air public channel, which is subject to active attacks. In order to reduce this cost, we introduce an *ephemeral channel*, which is a spread-spectrum channel temporarily established only for carrying the messages of a key exchange protocol.

The modified protocols are given in Fig. 17. In the first protocol, a message m containing a random number (less than 32 bits) for an ephemeral spread-spectrum channel, is appended to the first protocol message, so the message sent by A becomes (m, M_A). When B receives this message, he replies back on the (ephemeral) spread-spectrum channel generated by m, denoted as $Ch(m)$. Clearly, with this method only one message among M_A and M_B is carried over the open channel. A second modification to the protocol uses the same idea of sending a randomly generated ephemeral spreading code; however, this time it is sent as a single message over the open air channel. The protocol is described in

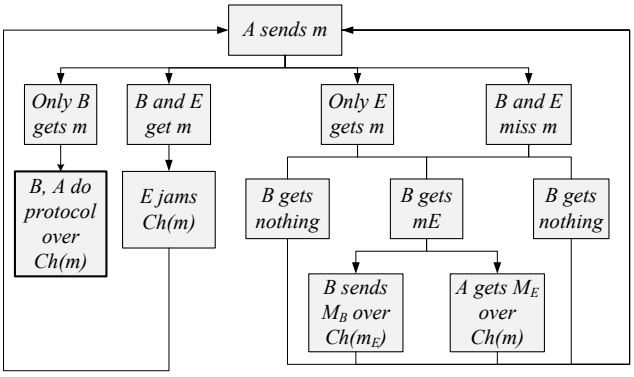


Figure 18: The flow chart for the key exchange described in Fig. 17 (right).

Fig. 17 (right). When B receives this message, he replies with his message M_B on channel $Ch(m)$, and similarly A replies back with M_A on the same channel. In other words, the whole key exchange protocol is moved to a spread-spectrum channel. Note that, both these methods are *general* modifications that can be applied to any key exchange protocol in Table 3.

These two approaches aim to carry at least some part of the messages on an ephemeral spread-spectrum channel to avoid signal jamming; however, this benefit comes at the expense that E could eavesdrop the open, short spreading code message and thus learn the channel $Ch(m)$, in which case she can successfully jam the message transfers and the benefit is lost. Clearly, the protocol may require an increased number of retransmissions for the message in Step 1, but the cost of these trials will be reasonable as the message containing the ephemeral channel code is typically short compared to the messages in the key exchange protocols, hence the increase in message sizes is negligible. The total cost of key exchange is improved if the benefit obtained in subsequent steps offsets the added cost incurred for transmission of the ephemeral channel code.

The idea of sending a random code for establishing an ephemeral channel is motivated by the fact that the wireless channel quality is random and time-varying, and packet losses are inevitable for any receiver including an attacker. We demonstrate the idea and its benefits through performance analysis based on the Rayleigh fading model as described in Section 2.2, which assumes sufficient fading diversity over the re-trials of Step 1. Step 1 can be carried out by (slow) frequency hopping over narrowband channels from one trial to the next, where different trials will experience independent fading (see Fig. 1). In other words, the nodes essentially keep "probing" the whole bandwidth for a good pair of fades. For example, future JTRS radios will have large bandwidths (multiple GHz), and even taking frequency coherence to be 100's of KHz there will be a large number of narrowband channels with independent fading which can be exploited for Step 1.

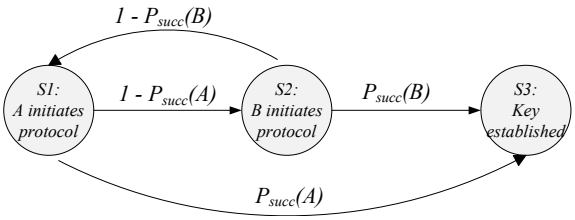


Figure 19: Markov state diagram for role-switching protocols.

6.2. Cooperative Jamming and Role-Switching

When the modified protocols with ephemeral spreading codes are employed, an eavesdropper near A will receive packets with a much lower probability of error than B, and thus it will take significant time for B to receive a packet that E does not. Since E's location is unknown, the system is dominated by the concern of a near eavesdropper. We address this by utilizing a physical layer technique referred to as *cooperative jamming* [25], [26], where a second antenna on the transmitter (or a collaborator) generates noise to reduce the signal-to-noise ratio at E. Although this will also degrade the signal for B, it can be shown that the overall impact is an improvement in the probability that "B receives the packet and E misses it". The total transmit energy is shared between the message signal and the (cooperative) jamming signal sent simultaneously.

Compared to the classical protocols, the performance of the modified protocols depends more strongly on which of the two parties initiates the protocol by sending the ephemeral code. Since nodes in general are not able to gauge their channel qualities to the other parties, we propose *role-switching* where nodes take turns initiating the protocol. For example, consider the modified protocol in Fig. 17 (right). Suppose A sends the message m, and starts listening for a reply on $Ch(m)$. If she cannot receive a message, she reverts to listening for a message from B over open air. In the meantime, B has either missed the message m, or his reply over $Ch(m)$ has not been delivered and so he could not receive a message from A. In either case, B realizes this and switches roles, and becomes the sender of the ephemeral code, m, over the open air.

6.3. Cost Evaluation and Numerical Results

The flow chart for the modified protocol (Fig. 17 (right)) is given in Fig. 18. A very similar flow chart for the protocol in Fig. 17 (left) is omitted. The analysis is similar to previous sections and thus the details are omitted for brevity. However, one important difference here is the consideration of the role-switching described above, as the cost calculation becomes complicated when nodes take turns in initiating the protocol. For that, the SFG based on the Markov state diagram in Fig. 19 is used to find the overall cost with role-switching, where the transfer functions on each branch (not given in the figure)

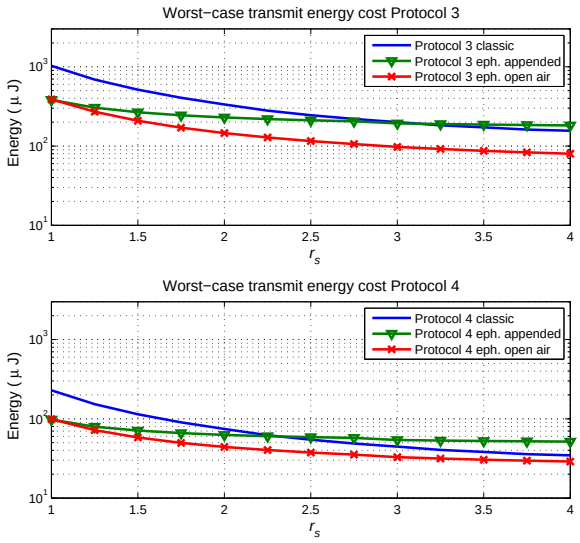


Figure 20: Cost values for Protocols 3 and 4 are plotted including their modified counterparts. The cost metric is the total transmit energy spent by A and B. For each protocol, three curves are given. The first curve is the classic protocol, the other two curves are for the modified protocols, where an ephemeral spreading code is added. Role-switching is assumed for calculating all of the curves as it also benefits the classical protocol. Cooperative jamming is employed only for the modified protocols. Plots for Protocols 1 and 2 show very similar behavior and are omitted here. The protocols with "ephemeral over open air" always requires less transmit energy on average compared to classic, although with savings reduced for less intense jamming. However, the method with "ephemeral code appended" starts to become more costly than classic after a certain value of r_s.

is found from the individual SFGs drawn for A and B based on the their flow charts as in Fig. 18.

The expected transmit energy cost for the modified protocols is shown in Fig. 20. For each protocol, we have three plots showing the cost for *Classic*, *ephemeral appended* and *ephemeral on open air*. As can be seen, the modified methods provide important performance gains compared to the classic protocols. These plots show that adding the ephemeral spreading code along with cooperative jamming and role-switching has a significant positive effect on performance. This demonstrates the value of integrating physical layer considerations into the protocol design.

6.4. Discussion

The modified protocols have been evaluated with the attacker model described in Section 2, as done for the classical protocols. However, it is also important to note how the modified protocols perform under different scenarios. For example, as opposed to the full-duplex attacker assumption, consider an attacker who has to choose between listening and transmitting at a given time. In order to eavesdrop the message m in Step 1, a half-duplex attacker

would need to keep silent, thereby increasing the chances B gets the channel code. Moreover, whenever E gets m in Step 1, she is forced to jam the channel $Ch(m)$ and spend energy (since over unjammed $Ch(m)$, protocol messages are received with probability one as opposed to the case with classical protocols), which could be advantageous in the case of an attacker who tries to save energy (e.g., battery-operated). In another scenario, the attacker may not constantly transmit or may even not exist for some time intervals. Whenever jamming stops, Step 1 completes quickly with little cost. Therefore, this protocol is robust in the sense that an attacker cannot force the system to switch to a high-cost solution with minimal effort, simply by signaling its existence.

7. Conclusions

We have presented a method for analysing the performance of cryptographic protocols on wireless channels in the presence of active adversaries, in which protocols are modeled as dynamic, probabilistic systems. We have demonstrated the utility of the method by applying it to the evaluation of key exchange protocols. The analysis leads to counter-intuitive results not suggested by prior approaches. It also led us to a novel approach to the design of key exchange protocols specifically tailored to the wireless environment. When evaluated using our method, this approach has better performance than traditional key exchange protocols. This shows the value of adopting a design approach that integrates physical layer features with traditional key exchange primitives.

For reasons of clarity of presentation, we have focused on communication cost as the principal metric in this paper. It will be evident that our method can also be used to study computation costs, protocol execution times, or other metrics of practical relevance. At the same time, it can be applied to compute moments and tail probabilities for these metrics. Further, its application is not limited to key exchange protocols, but can also be extended to study more complex classes of protocols, such as protocols for public key management, secure routing protocols for ad hoc networks, or protocols for secure distributed computing. We plan to explore these topics in our future work.

Acknowledgements

The research in this paper was sponsored in part by the U.S. Army Research Laboratory and the U.K. Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001, and by the National Science Foundation under grants CNS-0905349 and CNS-1018464. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

[1] R. E. Ziemer, R. L. Peterson, D. E. Borth, Introduction to Spread Spectrum Communications, Prentice Hall, 1995.

[2] C. J. Mitchell, F. Piper, Key storage in secure networks, Disc. App. Math. 21 (3) (1988) 215–228.

[3] L. Eschenauer, V. D. Gligor, A key-management scheme for distributed sensor networks, in: V. Atluri (Ed.), CCS, ACM, 2002, pp. 41–47.

[4] C. Boyd, A. Mathuria, Protocols for Authentication and Key Establishment, Springer-Verlag, 2003.

[5] M. Strasser, S. Capkun, C. Popper, M. Cagalj, Jamming-resistant key establishment using uncoordinated frequency hopping, in: IEEE Symposium on Security and Privacy, 2008, pp. 64 –78. doi:10.1109/SP.2008.9.

[6] R. Sakai, K. Ohgishi, M. Kasahara., Cryptosystems based on pairing., in: SCIS, 2000.

[7] R. Sittler, Systems analysis of discrete Markov processes, IRE Trans. Circuit Theory 3 (4) (1956) 257 – 266. doi:10.1109/TCT.1956.1086324.

[8] R. Howard, Dynamic Probabilistic Systems, J. Wiley & Sons Inc, 1971.

[9] D.-L. Lu, J.-F. Chang, Analysis of ARQ protocols via signal flow graphs, IEEE Trans. Comm. 37 (1989) 245 –251. doi:10.1109/26.20098.

[10] H. Zhai, Y. Kwon, Y. Fang, Performance analysis of IEEE 802.11 MAC protocols in wireless LANs, Wireless Communications and Mobile Computing 4 (2004) 917–931. doi:10.1002/wcm.263.

[11] T. Wan, A. Sheikh, Performance and stability analysis of buffered slotted ALOHA protocols using tagged user approach, IEEE Trans. on Vehicular Technology 49 (2000) 582 –593. doi:10.1109/25.832990.

[12] N. Potlapally, S. Ravi, A. Raghunathan, N. Jha, A study of the energy consumption characteristics of cryptographic algorithms and security protocols, IEEE Trans. Mob. Comp. 5 (2006) 128 – 143. doi:10.1109/TMC.2006.16.

[13] G. de Meulenaer, F. Gosset, F.-X. Standaert, O. Pereira, On the energy cost of communication and cryptography in wireless sensor networks, in: WIMOB, 2008. doi:10.1109/WiMob.2008.16.

[14] V. Shmatikov, Probabilistic analysis of an anonymity system., Journal of Computer Security 12 (3/4) (2004) 355 – 377.

[15] D. Guo, F. DiCesare, M. Zhou, A moment generating function based approach for evaluating extended stochastic Petri nets, IEEE Trans. Auto. Cont. 38 (2) (1993) 321 –327. doi:10.1109/9.250484.

[16] C. Baier, B. R. Haverkort, H. Hermanns, J.-P. Katoen, Performance evaluation and model checking join forces, Commun. ACM 53 (2010) 76–85.

[17] C. Bodei, M. Curti, P. Degano, C. Priami, A quantitative study of two attacks, Electron. Notes Theor. Comput. Sci. 121 (2005) 65–85.

[18] Y. Liu, P. Ning, H. Dai, A. Liu, Randomized differential DSSS: Jamming-resistant wireless broadcast communication, in: INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1 –9. doi:10.1109/INFCOM.2010.5462156.

[19] M. Strasser, C. Pöpper, S. Čapkun, Efficient uncoordinated FHSS anti-jamming communication, in: MobiHoc '09, 2009, pp. 207–218. doi:http://doi.acm.org/10.1145/1530748.1530778.

[20] M. Bloch, J. Barros, M. Rodrigues, S. McLaughlin, Wireless information-theoretic security, Information Theory, IEEE Transactions on 54 (6) (2008) 2515–2534. doi:10.1109/TIT.2008.921908.

[21] S. Xiao, W. Gong, D. Towsley, Secure wireless communication with dynamic secrets, in: INFOCOM 2010, IEEE, 2010.

[22] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, B. Yener, Robust key generation from signal envelopes in wireless networks, in: CCS, 2007, pp. 401–410.

[23] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, S. V. Krishnamurthy, On the effectiveness of secret key extraction from wireless signal strength in real environments, in: MobiCom, 2009, pp. 321–332.

[24] M. A. Zafer, D. Agrawal, M. Srivatsa, Limitations of generating a secret key using wireless fading under active adversary, IEEE/ACM Transactions on Networking PP (99) (2012) 1. doi:10.1109/TNET.2012.2183146.

[25] X. He, A. Yener, Cooperative jamming: The tale of friendly interference for secrecy, in: R. Liu, W. Trappe (Eds.), Securing Wireless Communications at the Physical Layer, Springer, 2010, pp. 65–88.

[26] S. Goel, R. Negi, Guaranteeing secrecy using artificial noise, Wireless Communications, IEEE Transactions on 7 (6) (2008) 2180–2189. doi:10.1109/TWC.2008.060848.

[27] D. Tse, P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, 2005.

[28] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, S. V. Krishnamurthy, On the effectiveness of secret key extraction from wireless signal strength in real environments, in: MobiCom, 2009, pp. 321–332.

[29] J. Hampton, M. Cruz, N. Merheb, A. Hammons, D. Paunil, F. Ouyang, MIMO channel measurements for urban military applications, in: MILCOM 2008, 2008, pp. 1 –7. doi:10.1109/MILCOM.2008.4753526.

[30] R. Poisel, Modern Communications Jamming Principles and Techniques, Artech House, 2011.

[31] J. Cavers, Mobile Channel Characteristics, Springer, 2000.

[32] J. Proakis, Digital Communications, 4th Edition, McGraw-Hill Science/Engineering/Math, 2000.

[33] R. Motwani, P. Raghavan, Randomized algorithms, Cambridge University Press, 1995.

[34] A. Menezes, P. C. van Oorschot, S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[35] M. Bellare, P. Rogaway, Entity authentication and key distribution, in: CRYPTO 1993, Vol. 773, 1993, pp. 232–249.

[36] R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: EUROCRYPT 2001, Vol. 2045 of LNCS, 2001, pp. 453–474.

[37] M. Bellare, D. Pointcheval, P. Rogaway, Authenticated key exchange secure against dictionary attacks, in: EUROCRYPT 2000, Vol. 1807, 2000, pp. 139–155.

[38] D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing, in: ASIACRYPT 2001, Vol. 2248 of LNCS, 2001, pp. 514–532.

[39] P. S. L. M. Barreto, M. Naehrig, Pairing-friendly elliptic curves of prime order, in: Selected Areas in Cryptography 2005, Vol. 3897 of LNCS, 2005, pp. 319–331.

[40] P. S. L. M. Barreto, B. Libert, N. McCullagh, J.-J. Quisquater, Efficient and provably-secure identity-based signatures and signcryption from bilinear maps, in: ASIACRYPT 2005, Vol. 3788 of LNCS, 2005, pp. 515–532.

[41] L. Chen, C. Kudla, Identity based authenticated key agreement protocols from pairings, in: CSFW, IEEE Computer Society, 2003, pp. 219–233.

[42] L. Chen, Z. Cheng, N. P. Smart, Identity-based key agreement protocols from pairings, Int. J. Inf. Sec. 6 (4) (2007) 213–241.

[43] R. Dupont, A. Enge, Provably secure non-interactive key distribution based on pairings, Disc. App. Math. 154 (2) (2006) 270–276.

[44] K. G. Paterson, S. Srinivasan, On the relations between non-interactive key distribution, identity-based encryption and trapdoor discrete log groups, Des. Codes Cryptography 52 (2) (2009) 219–241.