Improving Wireless Security Through Network Diversity

Tao Ye, Darryl Veitch, Jean Bolot

ABSTRACT

Data confidentiality has been an ongoing challenge in wireless networks. Wireless channels are prone to passive sniffing attacks and mobile devices can be difficult to secure due to a lack of computing power and weak supporting encryption components. However, modern mobile devices often have multiple wireless interfaces with diverse channel capacity and security capabilities, which means that mobile transactions (involving at least one mobile device) can be carried out using several links. In this paper, we show that the availability of diverse, heterogenous links (physical or logical) between nodes in a network can be used to increase the confidentiality of the information transmitted between them, on top of the availability or strength of underlying encryption techniques. Specifically, we make two contributions. First, we introduce a new security model using multiple channels to transmit data securely, based on the idea of deliberate corruption and information reduction. Second, in an effort towards provable security, we analyze the security of our model in the wiretap channel framework of information theory and characterize the secrecy capacity of our system. We show that in an adverse environment, simply splitting traffic to a more secure channel can indeed achieve perfect secrecy.

1. INTRODUCTION

Mobile phones have become ubiquitous, reaching an estimated 3.3 billion, i.e. half of the planet's population, in November 2007, with several countries having penetration rates much higher than 100% [1]. They are also typically viewed by their owners as important as their wallets or their keys. Furthermore, they are expected to become an important or even the most important conduit not just for voice calls, but for Internet services in the future as well [2]. Not surprisingly, mobile phones, and mobile devices in general, have become a fundamental component of modern lives (both civilian and military) and economies. They provide a panoply of services, ranging from mobile search to banking, advertising, social interactions or a growing number of location-aware services, that are critical to companies, governments, families and individual users. Critical services are valuable, and as such they must rely on some underlying guarantees, in particular availability and security. We focus on security issues in this paper, and specifically on data confidentiality.

Data confidentiality is a key component of security solutions and infrastructure for mobile environments. Indeed, wireless networks are especially prone to threats such as as eavesdropping and copying. The typical approach to protection is through data encryption. However, many link layer encryption schemes known to be weak (A5 [3] in GSM, WEP [4] in WiFi home or small business networks) continue to be present as network components. Strong end-to-end encryption techniques (e.g.TLS [5]) at a high layer may seem to be the solution, but these are not always available (for example, many web sites do not support them), they might be too costly to deploy and maintain effectively (e.g., require a Public keys shorter than recommended), and they may be strong in name only, containing undiscovered weaknesses.

Our approach to this problem of confidentiality is based on the following observation. Modern wireless devices such as laptops and smart phones typically connect to the network using a rich and heterogeneous set of physical interfaces [6]. For example, a cell phone typically includes a cellular voice/data interface (such as CDMA or GSM), a Bluetooth interface, as well as possibly high-speed data interfaces such as EV-DO, HSDPA, WiFi or WiBro/WiMax. A laptop can use a single WiFi card to connect to multiple 802.11 networks using virtual channels [7]. With the increasing number of options for wireless broadband to the home, even access points themselves (for example femtocells) are equipped with both wired DSL or cable and multiple wireless broadband interfaces.

We show in this paper that the availability of diverse, heterogenous links (physical or logical) between nodes in a network can be used to increase the confidentiality of the information transmitted between them, on top of the availability or strength of underlying encryption techniques.

Our goal is to design an overlay system, with efficient algorithms as its components, which effectively uses diverse links to provide security at low energy cost. We achieve this by designing for both low computation cost and bandwidth consumption overhead.

A natural first reaction to the idea of trying to exploit multiple paths for security is that strong public key cryptography can already provide adequate protection with only a single path. We offer the following counter-arguments to this view:

(1) Our approach would further increase the security of a system already deploying a strong encryption scheme at

very low cost. This could be considered a valuable feature in environments, such as military or financial institutions. Furthermore, we note that strong encryption systems which are thought to be secure may in fact be compromised and include either software bugs or unknown backdoors (whether maliciously introduced or not). By using multiple links, we greatly increase the physical difficulty for an adversary to eavesdrop (for example, sniffing CDMA data is quite expensive) and simultaneously open up the possibility of providing additional security via the splitting process itself.

(2) The reality is that many cryptographic components are in use despite their vulnerabilites being known. For example, 10 years after the discovery of the weakness of the original 802.11 encryption standard, WEP, more than 50% of access points are still using it [8].

(3) Strong encryption is not always available. In particular many web services (such as web based mail, offered by major portals) do not support it. Thus, typical situations such as using a public WiFi connection to read web based mail can easily expose private data to anyone who can sniff the WiFi link.

(4) Strong encryption can be computationally intensive, making it a challenge for certain applications such as distributing MPEG videos [9] and on small wireless device implementations. By trading off computation and communication, a lightweight encoding and traffic splitting scheme can help such devices achieve data confidentiality in an computation-aware fashion at levels well beyond those available today.

The idea of utilizing multiple links for communication was first employed to benefit from the increase of end-to-end bandwidth. Here however, we use it primarily to improve network security, in particular data confidentiality. The mobile ad hoc network (MANET) area has seen a number of approaches using multipath routing to securely deliver messages divided into pieces, such as SPREAD [10] which uses Shamir's Threshold Secret Sharing [11], and also [12] [13] [14]. However, because MANET is heavily focused on delivery reliability, the methods used to transform and divide messages often increase data redundancy to combat path failures and other packet losses. This often imposes high network bandwidth overhead [10], or high encoding and decoding times [12]. In contrast, we position ourselves within today's wireless network infrastructure, characterised by a small number of available paths but quite reliable delivery. Some paths, such as the cellular wireless hops, can also be bandwidth constrained. Therefore we look for a more efficient transformation and division scheme that provides stronger security guarantees, together with low bandwidth overhead.

One such scheme is the All-or-Nothing Transformation (AoNT) [15] [16]. An AoNT transforms blocks of messages, such that without *all* the transformed blocks one cannot recover the original message. The authors of [17] prove that a type of sparse parity checking codes, *spc2*, has this property. However, although *spc2* has linear complexity, encoding can still be slow if efficient decoding is needed, motivating faster algorithms.

Although the problem of how to securely split traffic across multiple paths has been considered, it is clearly not solved, especially in the setting of today's practical wireless environment. The challenges we address are two-fold. First, efficient multi-channel encoding and decoding algorithms, both in terms of computation and bandwidth utilization efficiency, and second, ways to describe and quantify the security they afford.

Traditionally, information theory has been mainly applied to the physical layer of wireless networks to understand physical channel capacity. The traditional security paradigm is based on a computational complexity approach, whereby brute force approaches to deriving cryptographic keys are shown to be computationally equivalent to benchmark problems agreed to be 'hard'. This approach treats security as an independent component from the underlying communication. We instead treat security as an integral part of communication, using an information theoretic principle, Wyner's *wiretap channel* [18] as our foundation model. In a wiretap channel, Alice and Bob tries to communicate secrets through a shared main channel C_M, while an eavesdropper Eve observes the transmitted information through a degraded from of C_M known as the wiretapper's channel C_W. Alice can encode secret messages to Bob reliably, i.e., with small enough error, while providing no information to Eve. In this work we relate the problem of splitting traffic to the wiretap channel by applying the wiretap channel concept to higher layers in the communication stack, and use it to understand fundamental limits on secrecy capacity in the multi-channel context.

We propose a *Multichannel Encryption Overlay (MEO)*, shown in figure 1, to split data transfer over multiple channels in a way that increases confidentiality significantly at low computation cost. The overlay is not a new crypto-scheme in the usual sense. Instead, it builds on an existing base cipher in a modular way and is based on two ideas: that information removal or *corruption* can thwart decryption, and that information rate *reduction* can greatly increase cracking time for those attacks based on sniffing cipher-text. Specifically, we propose to first perform an encryption E_0, in a very general sense, on the source S to form S', then split S' onto two or more channels, in such a way that most of the traffic is carried by channel 1, and the rest, after some additional encryption, on channel 2. Channel 1 essentially carries a corrupted version of S', and traffic on channel 2 is low rate (highly corrupted), and encrypted. The split streams can be reassembled at a network proxy location, or via a suitable multi-path recombination service, before reaching the final destination.

Our contributions are three-fold: (i) we show how existing network diversity can be applied to solve realistic wireless network confidentiality problems; (ii) we propose an inexpensive overlay technique (the MEO) which can split traffic over multiple channels while simultaneously increasing confidentiality; (iii) we point the way to the use of information theoretic based security in the context of multiple channel data transmission, in particular we derive bounds on the security capacity of the MEO under some assumptions, using the notion of secrecy capacity derived for the wiretap channel.

Our solution is energy efficient, and so conserves the most precious resource on a mobile device – battery power. It is also

modular and can be used with existing network configurations.

2. RELATED WORK

Many traffic dispersion schemes have been proposed to either provide secure message transmission in completely untrustworthy networks (such as mobile ad hoc networks), or to provide additional security provisions that complement existing mechanisms. We focus on the former. A MANET is a self-configurable, self-organizing network, with each node functioning both as an end host and a router. It is often assumed to have a fast changing topology due to frequent node relocation. To address the confidential data transmission problem, many schemes have been proposed to divide messages and send the fragments along multiple node-disjoint paths to the destination. The use of multiple paths is to increase the difficulty for adversaries to physically eavesdrop.

As an example, [10] achieves this by using Shamir's Threshold Secret Sharing [11]. The Shamir scheme divides a message into N parts in such a way that if fewer than T parts are obtained one cannot recover any bit of the message, but full reconstruction is possible using any T parts. This t-out-of-n threshold secret sharing is quite bandwidth inefficient, multipling the original message size by the number of paths. In Rabin's information dispersal (ID) [19] (used in [12]) a file is broken up into n pieces, such that any m pieces can be used to reconstruct it, where $n > m > 0$. Unlike Shamir's secret sharing, ID does NOT guarantee that information is not revealed if less than m pieces are intercepted. Although ID imposes less network bandwidth overhead, it often requires $O(n^2)$ encoding and decoding times, with n being the number of fixed length pieces. Therefore, both schemes rely strongly on a statistical argument that a large number of MANET nodes will need to be compromised to provide security.

Rivest proposed the 'package transform' [15] based on an All-on-Nothing-Transform, which preprocesses plaintext that is already divided into blocks through a matrix transformation, before sending the transformed blocks to the (block) encryption process. The authors of [17] use the sparse parity-check (SPC) code *spc2* to achieve an All-on-Nothing-Transform. They propose to encrypt the small amount of symbols (4%) and transmit it on a separate secure channel, while transmitting the rest of the parity-check coded data in the clear. Although the security property of *spc2* is desirable, the transformation process is still too expensive, on the order of $O(dn)$, where n proportional to original content length and d is a constant. In practice d needs to be larger than 10 to facilitate efficient decoding. Information slicing [20] proposes to transmit messages in a MANET anonymously but confidentially without going through a third-party anonymizer. A similar AoNT scheme is used to divide the messages, but the focus is on anonymization.

There are other schemes which with interesting features but which are again too expensive, such as [14].

3. THE ENCRYPTION OVERLAY SCHEME

In this section we first describe the Multichannel Encryption Overlay (MEO) scheme in its simplest form and its basic properties. We ignore practical issues such as loss, as well as packet headers and other overheads, re-packetization costs, byte alignment and so on, in order to focus on the core features.

The MEO has inherent security features, based on certain assumptions, which we describe below. In section 4 we present a different and more formal way to evaluate the security it provides.

3.1 Overview

We begin by assuming that there exists a bit source S going to a destination. The source S first goes through an encryption scheme, E_0 in figure 1, which generate a stream S' of encrypted bits assembled into packets. Note that E_0 is an encryption in the most general sense; it can be, but is not limited to, a cryptographic cipher such as a stream cipher. We are principally motivated to design a scheme that can be performed at low computational cost to enable high data rates. We will explore the design space of E_0 in later sections.

The overlay scheme is packet based and splits the encrypted packet output from E_0 over two channels as follows. For each packet of S' we *corrupt* it in a fundamental way by extracting one or more bits. As shown in figure 1, the packets with bits removed form a stream O_1 which is sent out along channel 1 (nominally one with a higher bandwidth). The missing bits are grouped together into packets, which are then encrypted using an additional cipher E_2 (also nominally a stream cipher), to form a stream O_2 which is send out on a second channel (nominally a lower bandwidth channel). To decrypt the overlay the receiver must collect the packets from both channels and invert the above steps to recover S'. Thus conceptually the overlay sits between the encryption and decryption functions of the underlying cipher E_0. In terms of implementation the E_0 and the overlay may be closely linked, for example in a driver which communicates with multiple physical interfaces, however we do not consider those details here.

There are important practical reasons why the *augmented system*, E_0 plus the overlay, can be more challenging to crack than any encryption alone. The main practical reasons are as follows. Access to two channels is now needed, which may be difficult, especially when they are over separate physical infrastructures. Second, even assuming full access, packet matching and reassembly must be performed whereby (assuming that E_2 is broken) the extracted bits sent over the second channel are reinserted into the correct bit positions in the correct packets from channel 1. Together these constitute significant extra work for the adversary.

We now comment on the overall system security. The MEO improves security base on two principles:
1) **Corruption,** disabling cracking on channel 1, and
2) **Information rate reduction,** slowing cracking on channel 2.
First, the removal of bits from the packets of S' effectively corrupts them from the point of view of anyone listening only on channel 1. Now consider the second channel. Because only a few bits are extracted from each packet of S', the bit rate on this channel is much lower than that on channel 1. As a result there is far less raw information available on channel 2. Thus, if for example E_2 were the same cipher as E_0, and assuming that cracking is based on the number of sniffed packets, then it

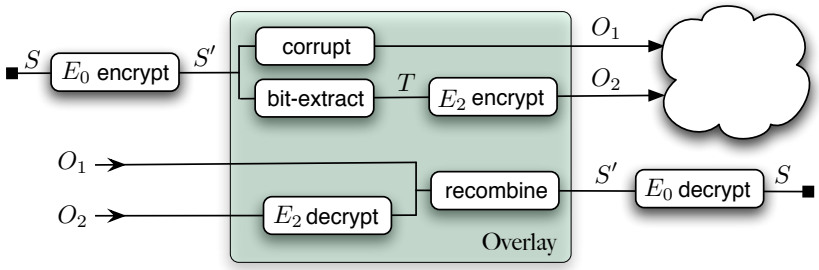

Figure 1: The Multichannel Encryption Overlay sits between the underlying encryption scheme E_0.

is much harder to break E_0 when observing channel 2 instead of S' directly. Of course, the negative consequence of this is an increase in latency, since packets on channel 1 cannot be decoded until the packet on channel 2 containing their missing bits arrives.

3.2 Analysis

3.2.1 Overlay Definition

We assume that there are two channels, labeled as $i = 1, 2$, over which the data is to be split. This can easily be generalized to more channels. Channel i is characterized simply by its capacity C_i bits per second (bps), and we will assume that $C_1 \geq C_2$.

The source traffic S is encrypted by some cipher E_0 to form the packet stream S' (see figure 1). The details of this cipher do not affect the overlay definition. Indeed it is a feature of the scheme that it can be used over different underlying single-channel ciphers in a modular way.

We will assume that S' can be viewed as a stationary packet process appearing over 'channel' $i = 0$, so that quantities such as its average packet arrival rate λ_0 pkts/sec, and data rate r_0 bps, are well defined. We make the simplifying (but not essential) assumption that all packets have the same size: p_0 bytes. Clearly $r_0 = 8p_0\lambda_0$. Similarly, λ_i, r_i and p_i can be defined for channels $i = 1, 2$.

Channel 1 For each packet of S', a bit-level corruption operator removes b bits, resulting in a smaller packet of $p_1 = p_0 - b/8$ bytes sent out into the packet stream O_1 along channel 1. As each packet of S' gives rise to a packet of O_1, clearly O_1 is stationary with $\lambda_1 = \lambda_0$ and rate $r_1 = (8p_0 - b)\lambda_0 = r_0 - b\lambda_0$.

The question of *which* bits are extracted does not affect our calculations here. We discuss this aspect in section 3.2.3.

Channel 2 Bits extracted from packets of S' are assembled into packets of size p_2 bytes to form the stream T. To hide these bits, T is encrypted using E_2 in a per-packet fashion and the resulting stationary stream O_2 sent out along channel 2. Since it takes $8p_2/b$ packets from S' to assemble a T packet, $\lambda_2 = b\lambda_0/8p_2$. Assuming for simplicity that E_2 does not alter packet size, we have $r_2 = 8p_2\lambda_2 = b\lambda_0 = br_0/8p_0$.

We see that, of the two overlay parameters b and p_2, the bandwidth sharing across the channels is controlled by b, since $r_1/r_2 = r_0/b\lambda_0 - 1$, whereas total offered load is invariant: $r_0 = r_1 + r_2$. The packet rate λ_2 is controlled by both parameters through the ratio p_2/b. This is therefore a key parameter for cracking time as explored below.

We now briefly consider the computational cost of the overlay, focussing on the client side (the other is similar). The cost can be expressed in terms of three main per-packet costs: x operations for the 'corruption' process to form a packet of O_1 from a packet of S', y operations for the bit extraction and repacketization operations to form a packet of T, and z operations for the encryption of a packet of T using E_2 to form a packet of O_2. The total cost per packet of S' is then $x + y + zb/8p_2$ operations.

The corruption cost x is likely to be small compared to the cost of E_0, however details will depend on the method by which bits are selected (see section 3.2.3). Since bit extraction and packetization are simple, y will be small. The cost z may be as large or larger than E_0, but since it does not apply to each packet in S' but only at the rate $\lambda_2 = b\lambda_0/8p_2$, its impact is small and controllable via b/p_2.

3.2.2 The Cracking Model

We wish to quantify the improvement in security obtained through using the overlay. To do so we need to provide a description of the mode of attack employed by an adversary together with a suitable *cracking model* describing the computation and/or time required for success. In the case of the overlay, we need a way to measure the increase in security it provides which is general enough to be meaningful regardless of the details of the underlying component ciphers E_0 and E_2, yet simple enough to be tractable.

We consider the class of ciphers which are vulnerable to attack based on intercepting (sniffing) cipher-text. Note however that even for ciphers outside this class (for example RSA where key factorization is the accepted mode of attack, a process which does not even look at cipher-text), corrupting the cipher-text may still render it difficult to recover the plain-text message, even with the key known.

Our cracking model can be described as follows. For ciphers in the above class, we quantify the success of the adversary via the notion of *the number of packets needed to recover a message*. As this number will vary depending on the specific message and other factors, we model it as a random variable

$N \geq 0$. Note that the distribution of N may depend on packet size and message length in complex ways and may be very difficult to derive explicitly. However, explicit knowledge is not necessary to analyze the impact of the overlay scheme. Using stationarity, from N we can define a 'cracking' time T simply as $T = N/\lambda$. For simplicity and to give quantitive results, we largely work with the average quantities $\mu_N = \mathbf{E}[N]$ and $\mu_T = \mathbf{E}[T] = \mathbf{E}[N]/\lambda$, but give the distribution in some cases.

Our use of N is consistent with Shannon's concept of the *equivocation measure*, which measures the level of uncertainly in our knowledge of the message (or key) as transmission proceeds. In Theorem 7 of Shannon's Communication Theory of Secrecy Systems [21], it is stated:

"*The equivocation of the first A letters of the message is a non-increasing function of the number N (letters) which have been intercepted.* "

The idea that knowledge of the key cannot decrease as transmission continues is consistent with our assumption that for a given message there exists a unique smallest number n of packets (a sample of the random variable N) needed to recover the message.

The full 'augmented' system, consisting of the overlay and the underlying cipher E_0, could be cracked in one of three ways:

(i) Cracking O_1 on channel 1 to S (channel 2 not needed)
(ii) Cracking O_2 on channel 2 to S (channel 1 not needed)
(iii) Cracking the overlay (cracking O_2 on channel 2 to T, successful packet matching and bit re-insertion into O_1 to recover S'), then cracking E_0 to S.

Of these, (ii) can be ignored because it is harder than (i), since with b small (in fact provided $b \leq 8p_2/2$), there is less information available in O_2 than in O_1, and in addition channel 2 is protected by E_2. We now consider (i) and (iii).

3.2.3 Impact of Corruption

We must first say more about how the bits for removal are chosen. We begin by considering the simple case where the bit *positions* are known to the adversary. Of course the bit *values* are not known as the bits are absent.

We consider that the adversary has already failed based on $n-1$ packets from O_1, and is now making an attempt based on n. Since S' is cipher-text, it is not apparent when a correct guess of bit values is made for any given packet. As a result, decryption must be attempted for each possible value to see if the guess was the right one. Thus the corruption will have the effect of multiplying the amount of computation by $(2^b)^n = 2^{bn}$, since, by the assumption above that the adversary cannot deduce the key based on $n-1$ packets, all bit values must be guessed correctly simultaneously over n packets.

For n large, the factor 2^{bn} represents a huge increase in cracking time. However, it is dwarfed by the effect of hiding the bit positions. If both the bit positions and values are unknown for each packet, the factor becomes

$$\left(\binom{8p_0}{b} 2^b \right)^n \quad . \tag{1}$$

For example, suppose the attacker has managed to sniff the first packet on channel 1 ($n = 1$), which has $p_1 = 240$ bytes, and that he knows that $b = 1$ were removed. The time the adversary takes to process this first packet (either by cracking, or by failing to crack), is magnified 3820 times (on average). If instead 3 bits were used, this becomes 9422443520 times. Failure to crack implies sniffing the next packet and starting from scratch with $n = 2$.

There exist ways in which the bit positions can be changed for each packet and yet be effectively hidden from the attacker. For example a pseudo-random sequence of high period could be used, the initializing seed and/or parameters of which could be communicated using a separate secure key exchange (for example RSA) prior to the data transfer. These operations must be factored into the cost of the overlay.

3.2.4 Impact of Information Reduction

Since the cracking time on channel 1 is likely to be exorbitantly high, for practical purposes S can only recovered by first cracking the overlay, beginning with channel 2. We start by comparing the average cracking time μ_T for the augmented system to μ_{T_0} for E_0 alone, based on this assumption.

The average time required to crack channel 2, that is to crack O_2 to T, is

$$\mu_{T_2} = \frac{\mu_{N_2}}{\lambda_2} = \frac{\mu_{N_2}}{\lambda_0} \cdot \frac{8p_2}{b} \tag{2}$$

which is proportional to $8p_2/b$. This ratio acts as a multiplier of the cracking time of E_2 arising directly from the reduction in the packet arrival rate. To facilitate comparison, assume that E_0 and E_2 are in the same class (for example, the same cipher but with different keys), so that $\mu_{N_0} = \mu_{N_2}$. The above equation then becomes

$$\mu_{T_2} = \mu_{T_0} \cdot \frac{8p_2}{b}. \tag{3}$$

We now consider the random variable T_2, the 'cracking time' on channel 2. Assume E_0 holds the same, T_2 is the additional time the overlay adds to the overall cracking time. Hence T_2 can be considered as a lower bound of the overlay cracking time, on top of existing security provided by E_0. By definition, $T_2 = \frac{N_2}{\lambda_2}$. The distribution function of T_2 is

$$F_{T_2}(x) = P\{T_2 \leq x\} = P\{\frac{N_2}{\lambda_2} \leq x\} = F_{N_2}(x\lambda_2) \tag{4}$$

Since E_0 and E_2 are in the same class, N_2 has the same distribution as N_0, $F_{N_2} = F_{N_0}$,

$$F_{T_2}(x) = F_{N_0}(x\lambda_0 \cdot \frac{b}{8p_2}) \tag{5}$$

Since we don't know the original distribution F_{N_0}, it is difficult to give a simple expression of the total cracking time (to recover the original message S) distribution of the overlay. In a simple illustration, the average time (ignoring packet matching and reassembly costs, and assuming the adversary can sniff both channels) is $\mu_T = \mu_{T_0} + \mu_{T_2} = \mu_{T_0}(1 + 8p_2/b)$, which is longer than the time μ_{T_0} using the single channel only by a

gain factor

$$r = \frac{\mu_T}{\mu_{T_0}} = 1 + \frac{8p_2}{b}. \quad (6)$$

It is not difficult to tune this gain to be significant. For example, assume $p_2 = 32$ bytes and $b = 1$ bit, yielding $r = 257$. The multichannel encryption takes 257 times longer to crack than the single channel alone! If it takes 5 hours to gather enough data to crack S', with the overlay this expands to 1285 hours. This will make 'drive-by' style WiFi spoofing for example much more difficult, as the adversary has to camp outside of your house for 53 days just to gather the raw data, instead of 5 hours.

We have shown that security can be greatly enhanced by using multiple channels, without introducing new encryption algorithms per se. We have however assumed that the adversary can only carry out *passive attacks*, that is that he cannot correspond actively to either of the communicating parties to carry out more specific plain-text or cipher-text attacks. Among passive attacks, we do not consider *side channel attacks*, which usually exploit a knowledge of timing or other information of one of the communicating parties. However, we expect that the multichannel nature of the overlay will greatly complicate many of these strategies as well.

3.3 Summary and WEP Example

Cracking the system via cracking channel 1 using brute force requires the adversary (assuming they know the value of b) to try all combinations of missing bits to undo the effect of corruption, resulting in a huge cost (equation (1)).

Cracking the overlay via first cracking E_2 on channel 2 (assuming both channels are eavesdropped and that cracking time is related to the number of packets sniffed) is slowed by the information reduction effect. When E_0 and E_2 are the same cipher, this multiplies the average cracking time by $r = 1 + \frac{8p_2}{b}$ (6).

Below we summarize a simulation study using WEP [22] to illustrate further the core corruption property of the MEO. We choose WEP here largely because of the ready availability of related software, and because it is well known to be weak. It is not intended to be the basis of any claim that corruption can *never* be corrected by some sufficiently determined adversary.

Our simulation is based on the two phased WEP cracking simulator developed by Bittau [8]. First, it generates all possible encrypted 'packets' (since in WEP cracking, only the fixed first two bytes of a packet are relevant, each encrypted 'packet' is only two bytes long). The number of different encrypted packets is decided by the size of the initiation vector. An initiation vector of 24bits will yield around 16 million different packets. These encrypted 'packets' correspond to the stream S'. Second, the simulator feeds this packet stream to the popular Aircrack (v2.41) [4] program to crack the WEP key using a software implementation of the well known weak IV attack [23]. This results in about 75% of keys being cracked, with 63% under 7 million packets. We set the cracking attempt timeout to (1,10) minutes, meaning we try to crack for 1 minute for every 100,000 packets below 3 million packets, and 10 minutes for every million packets above 3 million packets.

We next insert a bit removal 'corruption' phase between phases 1 and 2 above to simulate the channel 1 output O_1. The corruption is performed as shifting the encrypted packet payload b, $(b = 1, 2)$ bits to the left, starting at a random position, and pad on the right with zeros (or ones). With corruption added, the simulator failed to crack, that is empirically $\Pr(N = \infty) = 1$, in all cases. We tried extended cracking durations, (8,80) minutes, with the same result.

4. TOWARDS PROVABLE SECURITY

The approach typically taken in the literature to establish the value of a security scheme is to show that it is "hard to crack", which in practices amounts to showing that it is equivalent to a known computationally hard problem, such as the large prime factorization problem. A different approach, originated by Shannon [24] soon after he established the basis of information theory, is to estimate the intrinsic secrecy of the scheme using information theoretic concepts such as secrecy capacity.

We take this approach in the paper. We need to address two key questions, namely 1) how to model our overlay system using an information theoretic framework, and 2) how to derive or estimate the secrecy capacity of our proposed system. We address both questions in detail next.

4.1 Wyner's Wiretap Channel, Information Theory Framework

We set our security analysis in the information theoretic security framework, based on the wiretap channel [18]. The notion of wiretap channel was introduced by Wyner in 1975 [18], in which legitimate parties Alice and Bob are connected by a main channel and eavesdropper Eve has access to the communication through a degraded channel. Wyner showed there exists channel codes, without using any keys, that can guarantee a set degree of data confidentiality and are robust to errors on the main channel. Csiszár and Körner [25] characterized the secrecy capacity region of a more general case, in the broadcast channel. Our problem of removing bits from an encoded stream to achieve secure transmission, can be mapped easily into this framework. The traffic on any one channel can be seen as the wiretapper's channel while the whole traffic can be seen as the main channel. If an adversary can tap only one part of the communication but not all parts, with smart coding we can guarantee that the adversary cannot decode at all.

The wiretap channel model defines security in a completely different manner from the cryptography model. It seeks to combine source coding and channel coding techniques and take advantage of the errors of communication channels to guarantee that the mutual information rate between an eavesdropper and the original source is zero, i.e., the eavesdropper cannot decode the message. Recently we have seen much interests in the application of this information theoretic security definition. For example, [26] analyzed the application of sparse parity checking codes to the wiretap channel and calculated their secrecy capacity for an erasure wiretapper's channel. The authors made a fundamental connection between capacity achieving codes and its security features, and used this to guide the selection of codes. Recently, [27] [28] discussed the secrecy capacity of multiple antenna system, and quasi-

static Rayleigh fading channels, respectively, suggesting security can be provided at the physical level.

We now set up the analysis framework. In figure 2, C_M is the main channel, and C_W is the wiretapper's channel. We define an original message as a random variable V. After Alice encodes V, it becomes X. The main channel C_M is $X \rightarrow Y$, with X as a random variable denoting the input symbol to C_M and Y as the random variable denoting the output symbol of C_M. Bob decodes Y into V'. Similarly, the wiretapper's channel C_W is $X \rightarrow Z$. Assume we have a sequence of n input symbols, X^n, leading to Y^n and Z^n outputs.

We borrow the notation from [26] to express the security and reliability criterions:

$$P\{V \neq V'\} \rightarrow 0 \tag{7}$$

$$I(V;Z)/n \rightarrow 0, \text{ as } n \rightarrow \infty. \tag{8}$$

This states that codes have to satisfy two wiretap channel criterions. First the probability of decoding error on the main channel has to approach 0 asymptotically, second the mutual information rate between the wiretapper's input and the encoded message approaches 0 asymptotically.

The *secrecy capacity* is the maximum rate at which secure and reliable communication can happen between Alice and Bob, with zero leak to Eve. When C_M is less noisy than C_W, [25] shows that the *secrecy capacity*:

$$C_s = \max_{p(x)} [I(X;Y) - I(X;Z)]. \tag{9}$$

The secrecy capacity can be expressed in even simpler terms when special but common channels are used. When $I(X;Y)$ and $I(X;Z)$ can be individually maximized by the same $p(x)$ [29], the secrecy capacity is simply the difference in channel capacity:

$$C_s = \text{Capacity } (X \rightarrow Y) - \text{Capacity } (X \rightarrow Z). \tag{10}$$

Our problem can be studied in the following setup. Consider Channel 1 and Channel 2 together as a noiseless main channel (assume error free communication without splitting), while the information removal and either Channel 1 or Channel 2 can be modeled as the wiretapper's channel. This is shown in figure 3. We look for coding methods through this channel to force the mutual information rate between what is carried on channel 1, O_1, and the original source, S, to be zero. Since we control how S' is split over Channel 1 and Channel 2, we in fact design the type of channel for the wiretapper's channel.

An important aspect of design is understanding the fundamental limits of the channels. We are therefore interested in characterizing the secrecy capacity for our system. Since our main channel is noiseless, our wiretapper's channel is always more noisy. Equation 10 becomes

$$C_s = 1 - C_w. \tag{11}$$

4.2 Deletion Channel and its Capacity

In particular, our information removal scheme can be modeled as a type of binary deletion wiretapper's channel. An i.i.d. binary deletion channel is a binary channel where a bit stream passing through has each bit deleted with a probability d, independently. This differs from a binary erasure channel. In a binary erasure channel, each bit can be *erased* with some probability e, independently. When a bit is erased, its value is unknown but the position of the bit remains. For example, 1100100 might become 11?01?0. The deletion channel does not indicate the deleted bits' positions. For example, 1100100 can become 11010. Our proposed scheme, namely a Bit Removal Channel, is similar to the deletion channel characterized in [30] [31], because we delete bits from a stream in a random fashion. However, it is also different from the deletion channel, because we choose to randomly delete a fixed number of bits from each packet, as if dividing a bit stream into segments and imposing random deletion patterns onto each segment.

Assume our input is fixed length packets, and a fixed number of bits are deleted from the input, to result in fixed length packets output. The positions of the deleted bits are random. We model the deletion process as a binary deletion channel, where the input alphabet $\mathcal{A}_X$ is $\{0, 1\}$. The definition of our Bit Removal Channel (BRC) follows closely to [30], since their characterization makes use of block based deletion pattern with a random codebook in the simple decoding framework. This make it almost trivial for the BRC adaptation.

4.2.1 Definition and notations

- n: segment size.
- Input: $x^n = (x_1, x_2, ..., x_n)$ is a codeword where $x_i \in \mathcal{A}_X$. We denote X^n the input random variable. The alphabet size is 2.
- Deletion pattern: d^n is a binary vector $(d_1, d_2, ..., d_n)$, where $d_i = 1$ means the i-th symbol of x is deleted, and $d_i = 0$ means it is received. Let the total number of 0's in d^n be m.
- Output: $y^m = (y_1, ..., y_m)$, with $y_k = x_{i(k)}$, $1 \leq k \leq m$. Here $i(k)$ is the position of the k-th 0 in the sequence d. We denote Y^m the output random variable.
- θ: the percentage of bits that get through the channel. $\theta = \frac{m}{n}$.
- $H_0()$: binary entropy function, $H_0(x) = -x \log x - (1-x)\log(1-x)$
- R: A rate R is achievable if there exists a set of codes and a decoding rule that the average probability of decoding error tends to 0 as $n \rightarrow \infty$.

We are interested in the capacity of the BRC, through its association with the deletion channel. Although there has been recent advances in bounding the capacity of a deletion channel [32] [31] [30], there is still no single letter characterization of channel capacity. We then turn to the upper and lower bounds of this channel capacity.

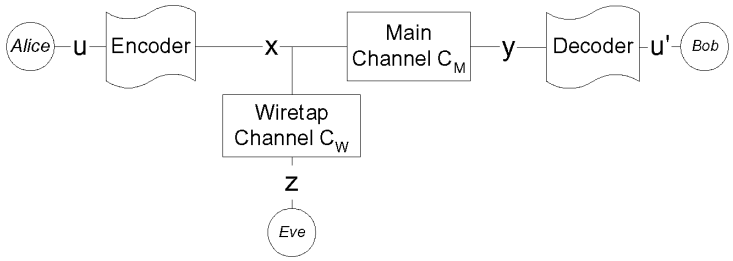

Figure 2: The Wiretap Channel Model. Alice tries to communicate secrets to Bob in the main channel, with Eve listening through a more noisy channel. We use coding to allow Bob to have error free decoding while Eve to gain no information.

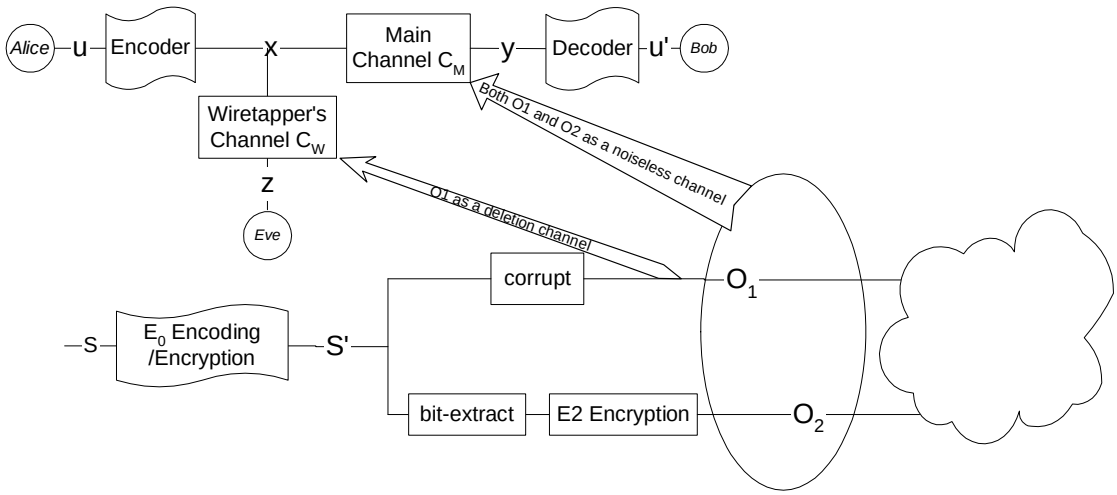

Figure 3: The MEO mapped onto the Wiretap Channel Model. Here we only show Channel 1 mapped onto the wiretapper's channel as a deletion channel when we study channel 1. The same can be done for Channel 2.

4.2.2 Lower Bound

Diggavi and Grossglauser in [30] used the random coding methods with simple decoding rules to derive the lower bound. We now summarize the basic method adapted to BRC below.

From [30], a ($\|\mathcal{C}\|$, n) code is a set of $\|\mathcal{C}\|$ codewords of length n that can encode a message by choose one of the codewords x. The number of codewords $\|\mathcal{C}\|$ relates to the achievable rate R in $\|\mathcal{C}\| = 2^{nR}$ when we generate a random codebook of 2^{nR} i.i.d. codewords. The deletion process transforms x to y where y is of length m. The decoding function tries to identify the a unique original x that contains the subsequence y. However, if there exists more than one codewords that contains subsequence y, a collision error occurs. The goal is to find the highest R such that the probability of collision error tends to 0. This is the lower bound of capacity.

They show that the average collision error probability per pair of codeword is only a function of n and m:

$$\bar{P} = \frac{F(n, m, 2)}{2^n}, \tag{12}$$

where F is function from common subsequences of random sequences results in the follow Lemma from [30]:

LEMMA 4.1. *For a given K-ary sequence y of length m, the number F(y, m, K) of K-ary sequences of length n which contain sequence y as a subsequence is given by:*

$$F(n, m, K) = \sum_{j=m}^{n} \binom{n}{j} (K-1)^{n-j}. \tag{13}$$

For our BRC, $K = 2$,

$$F(n, m, 2) = \sum_{j=m}^{n} \binom{n}{j}. \quad (14)$$

Hence, the pairwise error probability is bounded by:

$$\begin{aligned} \bar{P} &\leq \frac{1}{2^n} n \binom{n}{m} \quad (15) \\ &\leq \frac{1}{2^n} n 2^{nH_0(\frac{m}{n})} \\ &= n 2^{n(H_0(\theta)-1)}. \end{aligned}$$

Following the calculation of [30], summing pairwise error probability over all possible codewords x_1 (total number is 2^{nR}) bounds the error probability $\bar{P}_e$ as:

$$\bar{P}_e \leq 2^{nR} \mathbb{E}_{\mathcal{C}}[P\{\text{error}|x_1\}] = n 2^{n(R+H_0(\theta)-1)} \quad (16)$$

The error probability decreases exponentially with n when $R + H_0(\theta) - 1 < 0$, that is $R < 1 - H_0(\theta)$. Therefore there exists a deterministic codebook which has an achievable rate given by R, that decoding error approaches 0 asymptotically.

The above result only exists for $H_0(\theta) < 1$, that is when $\theta \geq 0.5$. To obtain a bound when $\theta < 0.5$ we turn to results of [31], where a simple lower bound for a general i.i.d. deletion channel is given. A general i.i.d deletion channel is different to the finite buffer model of [30] in that there is no block structure: each bit passing the channel simply has a deletion probability d. The simple lower bound is $C_{del} \geq A(1 - d)$, where $A = 0.1185$. The deletion probability relates to the θ parameter in the BRC as $\theta = 1 - d$ when n, the number of bits passing through the deletion channel, is large. In fact the i.i.d. deletion channel has a lower capacity than the BRC, because the BRC has side information, arising from the block (packet) structure, which tells us the exact proportion (θ) of bits remaining in each block. The lower bound for the general deletion channel is therefore also a lower bound for a BRC. It is looser than the previous one, but works for all θ values.

Finally then, a lower bound of the capacity of the binary deletion channel is given by:

$$C_{\text{BRC}} \geq \begin{cases} 1 - H_0(\theta), & \theta \geq 0.5 \\ A\theta, & \theta < 0.5. \end{cases} \quad (17)$$

4.2.3 Upper Bound

If the deletion patterns were communicated out of band (such as using sequence numbers), then the channel would be equivalent to an binary erasure channel, whose capacity is θ. Since conveying the deletion pattern constitutes side-information, this rate is an upper bound to this deletion channel capacity.

$$C_{\text{BRC}} \leq \theta. \quad (18)$$

4.3 (Expected) Secrecy Capacity of our MEO System

We now examine the question of the secrecy capacity of the system as a whole, and bring together the results of sections 3 and 4. Recall our mapping of MEO system to wiretap channel model in figure 3. Note that we can treat either channel 1 or channel 2 as a wiretapper's channel when calculating their secrecy capacity separately.

Section 3.2 provided a breakdown on the different pathways through which the system could be cracked. To apply the secrecy capacity results to such a breakdown we must first quantify the unknowns which are not provided by the secrecy capacity analysis. We do so with a simple model which captures the probabilities that Channel 1 or 2 might be eavesdropped on and/or cracked, as follows. We define

$$\begin{aligned} \text{Pr (channel 1 is sniffed)} &= q_1 \\ \text{Pr (channel 2 is sniffed)} &= q_2 \\ \text{Pr } (E_2 \text{ cracked} \mid \text{channel 2 sniffed}) &= q_{E_2}, \end{aligned}$$

We also assume independence between the corresponding events. The parameters q_1 and q_2 in particular allow us to take into account the relative difficulty of sniffing different physical interfaces. By controlling E_2 we can explore the impact of weakness in this part of the overlay.

Note that postulating the probability q_{E_2} that E_2 is cracked, although crude, is more general than the security analysis of section 3.2.4 in the sense that it is not based on any specific assumptions such as the amount of ciphertext that has been intercepted.

The cracking scenarios can be classified as:

1. Channel sniffed? 1-YES, 2-NO.
   Probability $p_1 = q_1(1 - q_2)$
2. Channel sniffed? 1-YES, 2-YES, E_2 cracked.
   Probability $p_2 = q_1 q_2 q_{E_2}$
3. Channel sniffed? 1-YES, 2-YES, E_2 not cracked.
   Probability $p_3 = q_1 q_2 (1 - q_{E_2})$
4. Channel sniffed? 1-NO, 2-YES, E_2 cracked.
   Probability $p_4 = (1 - q_1)(q_2) q_{E_2}$
5. Channel sniffed? 1-NO, 2-YES, E_2 not cracked.
   Probability $p_5 = (1 - q_1) q_2 (1 - q_{E_2})$
6. Channel sniffed? 1-NO, 2-NO
   Probability $p_6 = (1 - q_1)(1 - q_2)$.

The relevance of this breakdown is that a secrecy capacity can be readily determined for each case. In cases 5 and 6 the secrecy capacity is clearly equal to 1, since the overlay cannot be cracked by assumption. In case 2 it is equal to zero. Cases 1, 3 and 4 are non-trivial but can be treated using the secrecy results established earlier, because in each case the adversary eavesdrops only a single BRC. Specifically, in cases 1 and 3 the wiretapper's channel is channel 1 (stream O_1, see figure 1), a BRC with $\theta_1 = 1 - b/8p_0$. In case 4 the wiretapper's channel is channel 2 prior to encryption by E_2 (stream T from figure 1), a BRC with $\theta_2 = b/8p_0$. In each case, the (noise free) main channel corresponds to the (loss free) access to both channels 1 and 2.

We can now calculate the expected secrecy capacity of the system C_s by summing up the secrecy capacity of each case weighted by their probabilities. We obtain

$$\begin{aligned} C_s &= q_1(1-q_2)C_{s_1} + q_1 q_2 q_{E_2} \cdot 0 \\ &+ q_1 q_2 (1-q_{E_2}) C_{s_1} + q_2 q_{E_2}(1-q_1) C_{s_2} \\ &+ (1-q_1)(1-q_2)\cdot 1 + (1-q_1) q_2 (1-q_{E_2}) \cdot 1 \\ &= q_1(1-q_2 q_{E_2}) C_{s_1} + q_2 q_{E_2}(1-q_1) C_{s_2} \\ &+ (1-q_1)(1+q_2 q_{E_2}) \end{aligned} \tag{19}$$

where the component secrecy capacities C_{s_i} are given by

$$C_{s_i} = 1 - C_{\text{BRC}}(\theta_i)). \tag{20}$$

Although we do not have a single character expressions for the capacity of a BRC and therefore for C_s, equations (18) and (17) provide upper and lower bounds as follows

$$\begin{aligned} 1-\theta_i \le C_{s_i} \le H_0(\theta_i), \quad \theta_i \ge 0.5 \\ 1-\theta_i \le C_{s_i} \le 1 - A\theta_i, \quad \theta_i < 0.5. \end{aligned} \tag{21}$$

We can therefore derive bounds on C_s, the expected secrecy capacity of the system, by combining equations 21 and 19. Since we evaluate the bounds assuming Channel 1 has a larger capacity than channel 2, θ_1 is always greater than 0.5, while θ_2 is always less than 0.5. We obtain the following bounds:

$$\begin{aligned} C_s \ge\ & q_1(1-q_2 q_{E_2})(1-\theta_1) + q_2 q_{E_2}(1-q_1)(1-\theta_2) \\ & +(1-q_1)(1+q_2 q_{E_2}) \\ C_s \le\ & q_1(1-q_2 q_{E_2}) H_0(\theta_1) + q_2 q_{E_2}(1-q_1) A\theta_2 \\ & +(1-q_1)(1+q_2 q_{E_2}). \end{aligned}$$

We now evaluate the bounds as function of the system parameters q_1, q_2 and q_{E_2}, and the MEO parameter b via θ_1, and show the results in figure 4, and 5. In general the lower bounds decrease with θ_1 increases. When θ_1 is 0.5, i.e. traffic is evenly split between two channels, C_s is the highest.

In figure 4, we fix the probability of channel 2 being sniffed q_2 and the probability of E_2 being cracked q_{E2}, and plot three sets of secrecy capacity bounds based on different q_1 values. We can see that the secrecy capacity C_s decreases with the increase of q_1. It is especially interesting to see that even with $q_1 = 1$, i.e. channel 1 definitely sniffed and carry 90% of traffic, the secrecy capacity can still be at least 0.1, meaning there exists codes that can achieve perfect secrecy, although at a much lower communication rate. With higher q_1, C_s is more sensitive to the increase of θ_1 (steeper slope). In figure 5, we fix q_1, and plot two sets of secrecy capacity bounds based on different $q_2 q_{E2}$ values. The product of $q_2 q_{E2}$ is chosen as a variable because only when channel 2 is both sniffed and E_2 cracked, C_s is affected. We see that C_s only decreases slightly with a large jump of $q_2 q_{E2}$ from 0.0001 to 0.1. We limit the highest value of $q_2 q_{E2}$ to be 0.1 because following section 3.2.4, we assume q_{E2} is very small, i.e. the probability of cracking E_2 is small.

It is not surprising that secrecy capacity bounds is the most sensitive (decrease the most rapidly) when we vary q_1. The good news is, although in a wireless network we might not have control over q_1, we are usually aware of q_1, e.g. in an wifi hotspot, we can assume $q_1 = 1$. Then if we simply split

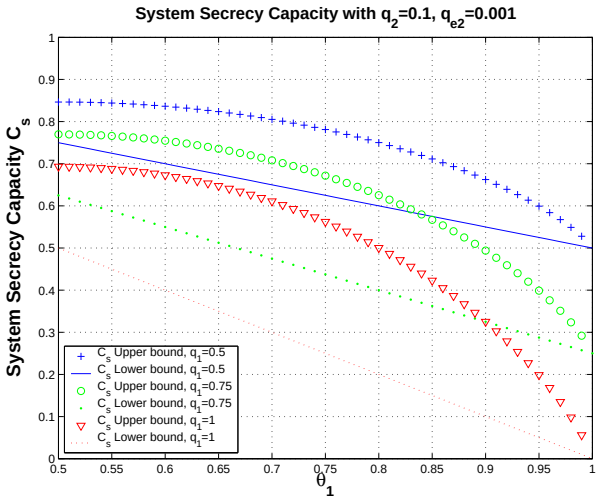

Figure 4: The expected system secrecy capacity with different values of q_1. The secrecy capacity C_s decreases with the increase of q_1. When $q_1 = 1$, i.e. channel 1 definitely sniffed and carry 90% of traffic, the secrecy capacity can still be at least 0.1, meaning there exists codes that can achieve perfect secrecy, although at a much lower communication rate. Among the three parameters, C_s is the most sensitive to changes in q_1.

just 10% of traffic to a more secure channel, with low q_2 and q_{E2}, there exist codes that can help achieve perfect secrecy. Further, the *more secure channel* can be either a strong encryption channel or a hard to sniff channel, and the more traffic we shift to this channel, the faster the rate (high C_s) is.

5. CONCLUSION

The main result of this paper is that a characteristic of many modern wireless devices, namely the availability of a rich and heterogeneous set of communication interfaces, can be used to increase the security of transactions carried out over those devices, on top of any confidentiality scheme existing in the devices. Thus, both a "secure" device implementing strong end to end encryption as well as a "weak" device implementing a known broken scheme such as WEP would benefit from our scheme. Our scheme is novel, based on the idea of deliberate corruption and information reduction and it is computationally lightweight, thus well adapted to battery-limited environments. Using tools a wiretap channel model in information theory, we show that derive bounds on our system's secrecy capacity and show that positive secrecy capacity is achievable even with minimal traffic splitting in practice.

We believe that our results open several interesting areas for future research, both of a practical and more theoretical nature. For example, how could knowledge about the timeliness of the information to be transmitted be exploited, for example in delay tolerant networks? Also, how would it be possible to take advantage of the mobility of users to spread

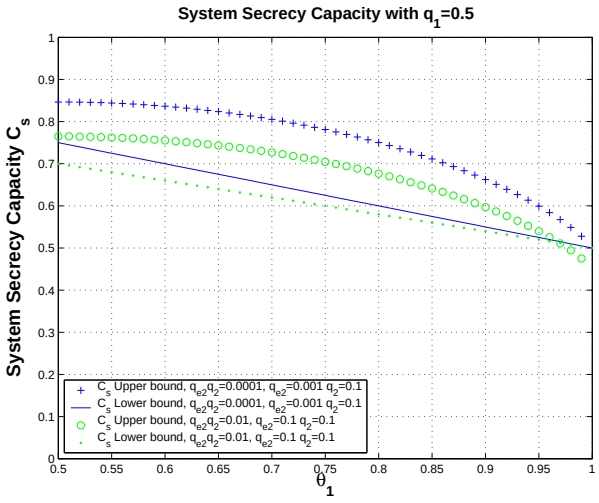

Figure 5: The expected system secrecy capacity with different values of $q_2 q_{E2}$. The secrecy capacity C_s is affected by only the product of q_2 and q_{E2}, since only when channel 2 is both sniffed and E_2 cracked, C_s is affected. The C_s value decreases only slightly when $q_2 q_{E2}$ increases from 0.0001 to 0.1.

streams of information not only between interfaces, but between users one might get in the neighborhood of (and rely on those users or their relays delivering a substream to the destination)? On a more theoretical level, what is the best way to combine encryption and compression to derive efficient confidentiality scheme? And finally, and possibly most challenging, is it possible to derive the secrecy capacity (not bounds) of general, practical scheme – and can we derive "composition laws" that would deliver the secrecy capacity of a system given the secrecy capacities of the various system components? We intend to investigate several of these topics in the future.

6. REFERENCES

[1] wikipedia.org, "Mobile phone."

[2] S. Keshav, "Why cell phones will dominate the future Internet," *Computer Communications Review*, vol. 35, no. 2, 2005.

[3] E. Barkan, E. Biham, and N. Keller, "Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communications," vol. 2729, pp. 600–616, 2003.

[4] C. Devine, "Aircrack-2.41," 2004. [Online]. Available: http://aircrack-ng.org/

[5] T. Dierks and E. Rescorla, "IETF RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1," *IETF*, April 2006.

[6] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee, "Mar: A commuter router infrastructure for the mobile internet," in *ACM Mobisys*, 2004.

[7] R. Chandra, P. Bahl, and P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card," in *Proceedings of IEEE Infocom 2004*, Hong Kong, March 2004.

[8] A. Bittau, M. Handley, and J. Lackey, "The Final Nail in WEP's Coffin," in *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 386–400.

[9] B. M. Macq and J.-J. Quisquater, "Cryptology for digital tv broadcasting," *Proceedings of the IEEE*, vol. 83(6), pp. 944–957, 1995.

[10] W. Lou, W. Liu, and Y. Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks," in *IEEE Infocom*, 2004.

[11] A. Shamir, "How to share a secret," pp. 612–613, 1979.

[12] P. Papadimitratos and Z. Hass, "Secure Data Transmission in Mobile Ad Hoc Networks," in *Wireless Security (Wise) Workshop at Mobicom*, 2003.

[13] E. Ayanoglu, I. C.L., R. Gitlin, and M. J.E., "Diversity coding for transparent self-healing and fault-tolerant communication networks," pp. 1677–1686, 1993.

[14] R. Vasudevan and S. Sanyal, "A Novel Multipath Approach to Security in Mobile Ad Hoc Networks," in *International Conference Computers and Devices for Communication (CODEC)*, Kolkata, India, 2004.

[15] R. L. Rivest, "All-or-nothing encryption and the package transform," in *the 1997 Fast Software Encryption Conference*, 1997.

[16] D. Stinson, "Something about all or nothing (transform)," 1999.

[17] J. Byers, M. C. Cheng, J. Considine, and G. Itkis, "Securing Bulk Content Almost for Free," 2004.

[18] A. D. Wyner, "The wiretap channel," *Bell. System Tech Journal*, vol. 54, pp. 1355–1387, 1975.

[19] M. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," pp. 335–348, 1989.

[20] S. Katti, J. Cohen, and D. Katabi, "Information Slicing: Anonymity Using Unreliable Overlays," in *Usenix NSDI*, 2007.

[21] C. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, vol. 28(4), pp. 656–715, 1949.

[22] "ANSI/IEEE Standard 802.11: Wireless LAN Medum Access Contral and Physical Layer (PHY) Specifications," *IEEE Computer Society*, 1999.

[23] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," vol. 2259, pp. 1–24, 2001.

[24] C. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*.

[25] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *IEEE Transactions on Information Theory*, vol. 24, pp. 339–348, May 1978.

[26] A. Thangaraj, A. R. Calderbank, and J.-M. Merolla, "Applications of LDPC codes to the Wiretap channel," *Draft*, 2007.

[27] A. Khisti and G. Wornell, "Secure transmission with multiple antennas: The MISOME Wiretap Channel," *ArXiv e-prints*, vol. 708, Aug. 2007. [Online]. Available: http://adsabs.harvard.edu/abs/2007arXiv0708.4219K

[28] J. barros and M. R. D. Rodrigues, "Secrecy capacity of wireless channels," *ISIT*, 2006.

[29] M. van Dijk, "On a special class of broadcast channels with confidential messages," *IEEE Transactions on Information Theory*, vol. 43, pp. 712–714, Mar. 1997.

[30] S. Diggavi and M. Grossglauser, "On information transmission over a finite buffer channel," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1226–1237, March 2006.

[31] M. Mitzenmacher and E. Drinea, "A simple lower bound for the capacity of deletion channels," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4657–4660, October 2006.

[32] E. Drinea and M. Mitzenmacher, "On lower bounds for the capacity of deletion channels," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4648–4657, October 2006.