Cognitive Radio from Hell: Flipping Attack on Direct-Sequence Spread Spectrum

J. Harshan† and Yih-Chun Hu*
†Indian Institute of Technology Delhi, India, *University of Illinois Urbana-Champaign, USA
Email: jharshan@ee.iitd.ac.in, yihchun@illinois.edu

***Abstract*—In this paper, we introduce a strong adversarial attack, referred to as the flipping attack, on Direct-Sequence Spread Spectrum (DSSS) systems. In this attack, the attacker, which is appropriately positioned between the transmitter and the receiver, instantaneously flips the transmitted symbols *in the air* at 50% rate, thereby driving the channel capacity to zero. Unlike the traditional jamming attack, this attack, when perfectly executed, cannot be detected at the receiver using signal-to-noise-ratio measurements. However, this attack necessitates the attacker to perfectly know the realizations of all the channels in the model. We first introduce the consequences of the flipping attack on narrowband frequency-flat channels, and subsequently discuss its feasibility in wideband frequency-selective channels. From the legitimate users' perspective, we present a method to detect this attack and also propose heuristics to improve the error performance under the attack. We emphasize that future cyber-physical systems that employ DSSS should design transceivers to detect the proposed flipping attack, and then apply appropriate countermeasures.**

I. Introduction

With wireless communication being an integral part of most cyber-physical systems [1], [2], e.g. urban transportation, smart-grid and other IOT systems, it is imperative to not just secure wireless links from external attacks such as jamming, but also envision new attacks and provide suitable countermeasures against them. In this paper, we are interested in external attacks that can drive the channel capacity of communication between two legitimate users to zero. Although, jamming is a straightforward way of realizing such an objective, such attacks can be detected at the legitimate users by measuring their signal-to-interference-noise-ratio (SINR). In other words, jamming is not a stealth attack. From the jammer's perspective, while attack detection is a disadvantage, the jammer need not know the wireless channel between the source and the destination (other than the band of communication). In this paper, we would like to ask a converse question: *Given perfect knowledge of the wireless channel between the source and the destination, is it possible for a sophisticated external attacker to drive their channel capacity to zero in stealth?*

To answer the above question, we envisage sophisticated threat models arising out of full-duplex radios [3] that operate as hidden relays in between a source and a destination. Loosely speaking, this threat comes under the well-known framework of *man-in-the-middle attacks*, wherein the attacker can manipulate the transmitted symbols before they reach the destination. Although instantaneous modification of transmitted symbols has been addressed in theory to mitigate interference in wireless networks [4], [5], [6], this has not been studied as a threat to wireless security. We propose a new adversarial attack on wireless communication between two legitimate users, wherein the attacker, who is appropriately positioned between the two users, manipulates the symbols *in the air*. Specifically, in the case of Binary Phase Shift Keying (BPSK), the attacker flips the BPSK symbols at 50% rate independently, thereby driving the mutual information of the channel to zero. We refer to such an attacker as Cognitive Radio from Hell (CRFH), wherein the phrase *from hell* is used to highlight the legitimate users' inability to avoid this attack.

We first apply the proposed attack on frequency-flat narrowband communication channels, and subsequently discuss its impact on frequency-selective wideband communication channels. Due to practical constraints on applying this attack on wideband channels, we discuss a variant of the attack wherein the transmitted symbols on the delayed paths are manipulated, while keeping the symbol on the first significant path untouched. When perfectly executed, this attack can force the destination to combine the observations from all the paths, thereby degrading the error performance. From the legitimate users' perspective, we discuss methods to detect this attack and also propose heuristics to improve the error performance under the attack. Throughout the paper, we refer to the source, the destination, and the attacker as Alice, Bob, and Eve, respectively.

II. Flipping Attack on Narrowband Channels

Consider a narrowband communication channel between two legitimate players Alice and Bob (each equipped with single antenna), characterized by the signal model

$$y_k = \sqrt{P}hx_k + n_k, \tag{1}$$

where $y_k \in \mathbb{C}$ is the received symbol by Bob at the k-th time-instant, $x_k \in \{-1, +1\}$ is the BPSK symbol[1] transmitted by Alice, $n_k \in \mathbb{C}$ is the additive white Gaussian noise (AWGN) distributed as $\mathcal{CN}(0, \sigma^2)$, and $h \in \mathbb{C}$ represents the fading coefficient distributed as $\mathcal{CN}(0, 1)$. The average signal-to-noise ratio (SNR) of this channel is $\frac{P}{\sigma^2}$. We assume a quasi-static fading channel where the realization h is fixed over several consecutive symbol intervals. We denote the symbol period by T_s seconds, and use t_{main} to denote the time taken by the symbol to reach Bob. For the above described model, let

[1]We have used binary phase shift keying (BPSK) constellations for the sake of introducing the flipping attack. However, this attack can also be generalized to higher-order constellations.

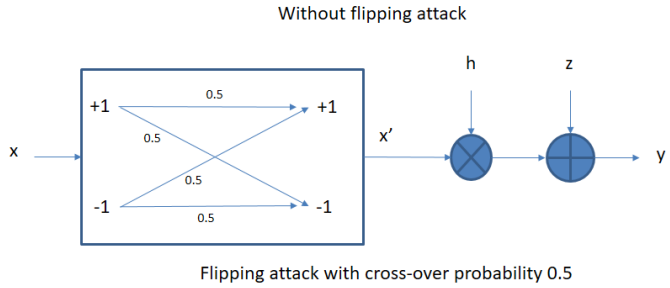

Fig. 1. In the flipping attack, the attacker instantaneously modifies the symbols in the air such that the receiver views the compound channel shown above. From information processing inequality, it is easy to verify that $I(x:y)=0$ with perfect execution of the flipping attack.

us imagine a sophisticated adversarial attack initiated by Eve, who is physically positioned somewhere between Alice and Bob. We envisage a powerful attack, referred to as the flipping attack, wherein Eve is able to receive the transmitted symbol from Alice, decode it, and then transmit a suitably modified version to Bob within the symbol period. With such a strong attack model, let the additional processing-delay introduced by Eve be t_p seconds, and the additional path-delay introduced by Eve be t_{side} seconds. With that, the modified symbol reaches Bob after $t_p + t_{side}$ seconds. Provided we have

$$t_{main} \leq t_p + t_{side} << t_{main} + T_s, \tag{2}$$

it is clear that Bob cannot resolve the transmitted symbol from Alice and the one manipulated by Eve. If we denote the manipulated symbol by $f(x,h,g)$, where g is the channel between Eve and Bob, then the received symbol after the attack is

$$y_k = \sqrt{P}hx_k + gf(x_k,h,g) + n_k. \tag{3}$$

In the flipping attack, Eve chooses the function $f(\cdot)$ such that $gf(x_k,h,g) = -2\sqrt{P}hx_k$, which implies that Eve has perfect knowledge of g and h. With such an "in the air" modification, Bob will receive a flipped version of the transmitted symbol, given by

$$y_k = -\sqrt{P}hx_k + n_k. \tag{4}$$

In the case of no attack, the received symbol is as given in (1). Note that if the attacker decides to flip the symbols at 50% rate independently, then the BPSK symbols would go though the effective channel as shown in Fig. 1. We can envision the attacker to flip the BPSK symbols by tossing a fair coin independently, thereby resulting in Bernoulli distribution with probability 0.5. The following proposition on the above attack is straightforward to prove.

Proposition 1: From the information processing inequality [7], the mutual information $I(x:y)$ of the compound channel shown in Fig. 1 is zero.

The condition in (2) indicates that both path-delay and processing-delay through Eve are bottlenecks for perfectly executing the flipping attack. In extreme narrow-band communication, i.e., when the symbol period T_s is much larger than the delay introduced by Eve, a perfectly executed flipping attack can drive the channel capacity to zero. However, in wideband communication, i.e., when T_s is extremely small relative to the effective delay introduced by Eve, the modified symbol $gf(x,h,g)$ is likely to reach Bob after the current symbol period. This implies that Bob will have to treat the delayed modified symbol from Eve as noise, which in turn will lower the signal-to-interference-noise ratio (SINR) of the next transmitted symbol. In such a case, although the delayed modified symbol degrades the error performance, this is not the intended consequence of the flipping attack. Recall that the primary objective of the flipping attack is to drive the channel capacity to zero in stealth.

III. Flipping Attack on Wideband Channels

It is clear that driving the channel capacity to zero through the flipping attack is challenging when the symbol period T_s is extremely small. However, a variant of this attack can be envisioned on a certain class of frequency-selective wideband channels, wherein multiple copies of the transmitted symbol arrive at Bob at different symbol periods; This way Eve gets sufficient time to execute the flipping attack on the delayed copies. Such channels are characterized by the signal model

$$y_k = \sum_{l=0}^{L_d-1} h_l x_{k-\tau(l)} + n_k, \tag{5}$$

where y_k denotes the baseband sample received by Bob at k-th symbol period, and $\{h_l \mid 0 \leq l \leq L_d-1\}$ denotes the set of fading coefficients associated with the delayed copies. Here $\tau(l)$ denotes the delay of l-th copy measured in terms of number of samples. We refer to each of these copies as a *tap*. Unlike in (1), we use $P=1$ in the signal model in (5). The practical constraints on executing the flipping attack forbids Eve from flipping the BPSK symbol arriving on the first significant tap, i.e., on h_0. However, it is reasonable to assume that Eve can flip the BPSK symbols arriving on subsequent taps as she gets relatively longer time for processing and forwarding the received signals. In this case, we assume that Eve has perfect knowledge of the power-delay profile (PDP) of Alice-Bob's channel, and also its realizations.

Henceforth, throughout the paper, (i) h_0 is referred to as the main tap, whereas $\{h_l \mid 1 \leq l \leq L_d-1\}$ are referred to as secondary taps, and (ii) flipping attack refers to the case when the symbol on the main tap is undisturbed, whereas the symbols on the secondary taps may be flipped independently at 50% rate. In practice, the feasibility of executing the flipping attack depends on PDP of Alice-Bob's channel, particularly the delay of the secondary taps with respect to the main tap. Note that the objective of the attacker is to make sure that signals received on the secondary taps carry no information about the transmitted symbol.

A. Flipping Attack

We assume that Alice and Bob communicate over a wideband channel using a DSSS system, wherein an N-length spreading code, which is securely shared between them, is applied on each of the BPSK symbols. We assume that Eve can accurately flip the chips with perfect knowledge of the

channel estimates of Alice-Bob, Alice-Eve, and Eve-Bob. With perfect attack, Eve flips the BPSK symbols at 50% rate. Note that once Eve decides to flip a BPSK symbol, she has to flip all the N chips associated with that symbol. At the destination, Bob uses RAKE receivers to resolve the symbols arriving on the L_d taps. After suitable correlation operation using the N-length spreading code on the received samples, Bob arrives at the L_d equivalent received symbols, given by

$$y_{k,l} = h_l x_k + z_{k,l},\ 0 \leq l \leq L_d - 1,$$

where the new subscript l is used to represent symbols received from the l-th finger, and $z_{k,l}$ is the effective AWGN, distributed as $\mathcal{CN}(0, \sigma^2)$, resulting from the correlation operation of the RAKE receiver. With that, the received symbols from the L_d fingers are of the form

$$\begin{aligned} y_{k,0} &= h_0 x_k + z_{k,0}, \\ y_{k,l} &= b_{k,l} h_l x_k + z_{k,l},\ 1 \leq l \leq L_d - 1, \end{aligned}$$

where the polarity of $b_{k,l} \in \{-1, 1\}$ depends on attacker's choice. With uncoded communication, the Maximum Likelihood (ML) decoder expression is given by

$$\tilde{x}_k = \arg min_{x \in \{-1,+1\}} \sum_{l=0}^{L_d-1} |y_{k,l} - h_l x|^2, \qquad (6)$$

where h_l is the perfect estimate of the channel on the l-th tap. It is clear from (6) that that attacker can force degraded error performance by flipping symbols on some of the taps. On the defensive side, the receiver Bob may choose to use only the main tap fearing that the secondary taps might have been flipped. In such a case, the decoding operation is given by

$$\tilde{x}_k = \arg min_{x \in \{-1,+1\}} |y_{k,0} - h_0 x|^2. \qquad (7)$$

Note that although the attacker's signals are not affecting the error performance using (7), the overall error performance will be worse than the no-attack case because Bob has not incorporated all the independent taps. In the next section, through simulations, we demonstrate the impact of the flipping attack on wideband frequency-selective channels with two taps and four taps in the delay-spread domain.

B. Simulation Results

In our setup, data communication between Alice and Bob takes place through a sequence of frames. Each frame constitutes 100 BPSK symbols, out of which 20 are occupied by the pilots. The pilot symbols, which are a priori fixed between Alice and Bob, also take values from BPSK constellation $\{-1, 1\}$. At Alice, a block of input bits of length 3968 bits are fed to a Rate-$\frac{1}{2}$ turbo encoder in feed-forward configuration [7 5], whose details are available in reference [8]. The corresponding output bits (7889 bits in number) are spread across the data part of several frames. Once the data and pilot symbols are packed, the frames are transmitted sequentially through a DSSS system, i.e., each BPSK symbol is multiplied by a spreading sequence of chip-rate $N = 128$. The frames are transmitted through the following wireless channels: (i) a two-tap channel with average power-profile $\{E\{|h_0|^2\}, E\{|h_1|^2\}\} = \{0.5, 0.5\}$, and (ii) a four-tap channel with average power-profile $\{E\{|h_0|^2\}, E\{|h_1|^2\}, E\{|h_2|^2\}, E\{|h_3|^2\}\} = \{0.4, 0.3, 0.2, 0.1\}$. We assume that each h_l is a circularly-symmetric complex Gaussian random variable, who realization remains fixed throughout the frame, and take independent realizations across frames.

Meanwhile, Bob uses RAKE receivers to resolve the dominant multipaths in the channel, and also estimates the channel realization on each tap using the pilot symbols. Subsequently, Bob combines the received symbols from all the taps to obtain log-likelihood ratio (LLR) on each BPSK symbol. Finally, the LLRs from each frame are forwarded to the turbo decoder, which processes them to decode the information bits. A total of 10 iterations is used for the message passing algorithm in the turbo decoder.

In Fig. 2, we plot the Bit Error Rate (BER) performance of the above discussed system on the two-tap channel in three scenarios: (i) without attack - Eve does not flip the symbols on any tap, and Bob combines the observations on both taps, (ii) with attack - Eve flips the symbols on the second tap at 50% rate independently, and Bob combines the observations on both taps, and finally (iii) with attack - Eve flips the symbols on the second tap at 50% rate independently, and Bob discards the symbols received on the second tap. The plots indicate that an attack-ignorant Bob will experience error-floor behaviour in BER by blindly combining the observations on both taps, whereas an attack-aware Bob can recover the bits with some SNR loss by discarding the observations from the secondary tap. Similar experiments are also repeated for the four-tap channel, and the corresponding BER results are presented in Fig. 3. In this case, we consider the following attack scenarios: (i) only the fourth tap is attacked, (ii) both the third and the fourth taps are attacked, (iii) second, third, and fourth taps are attacked. The plots in Fig. 3 show that Bob can avoid degraded error performance if he can somehow detect the attacked taps and discard them when computing the LLR values.

To obtain the simulation results in Fig. 2 and Fig. 3, we have assumed that Eve knows the locations of the pilot symbols, and therefore she does not flip the pilots. As a result, Bob is able to estimate the channel on each tap accurately. However, since Eve strategically flips only the data symbols at 50% rate, Bob is forced to combine all the taps as he is attack-ignorant. Thus, an important question to answer along this direction is *how can Bob identify an unreliable tap?* This question will be addressed in the next section.

C. Attack Detection Techniques

From Eve's perspective, a critical task is to attack only on the data symbols. When this is ensured, Bob cannot detect the attack. On the other hand, when Eve flips the pilot symbols as well, then Bob can detect this attack by observing the polarity of the received symbols on the pilot locations. Therefore, from the legitimate users' perspective, in order to detect the flipping attack they must obfuscate the location of pilot symbols in every frame so that Eve is forced to flip some of the pilot symbols. To achieve this, we enable Alice and Bob to share a secret key using which the random positions

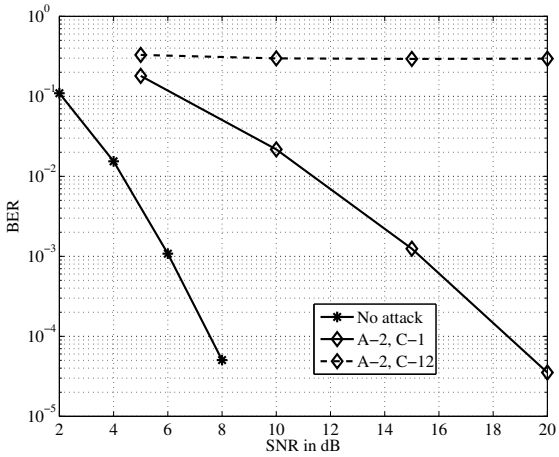

Fig. 2. Turbo-coded error performance on a two-tap channel with perfect estimate of Alice-Bob's channels at Eve. "A-2, C-1" implies that Eve flips the symbols on the second tap at 50% rate, while Bob combines the received symbols on first tap to obtain the LLR. Similarly, "A-2, C-12" implies that Eve flips the symbols on the second tap at 50% rate, while Bob combines the received symbols on the first and the second taps to obtain the LLRs. The plots show that combining the flipped symbols degrades the error performance.

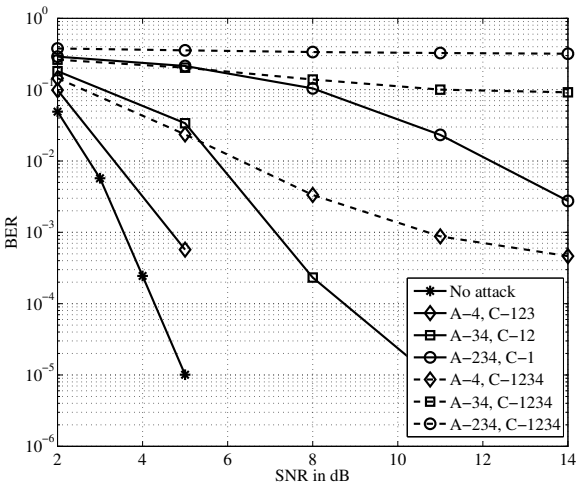

Fig. 3. Turbo-coded error performance on a four-tap channel with perfect estimate of Alice-Bob's channel at Eve. Notations in the legend are similar to those in Fig. 2. The plots show that Eve has to flip the secondary taps with significant energy to force degraded error performance at Bob.

of the pilots are determined. Since the positions of pilots are randomly chosen based on a pseudo-random generator, Eve cannot distinguish the pilots in the frame. Meanwhile, Bob's strategy is to observe the polarity of the received symbols on the pilot locations, and then detect the attack if at least one of the pilot symbols is flipped.

By observing the polarity of the received symbols on the pilot locations, there is a non-zero probability with which Bob fails to detect the attack. At high SNR values, this event corresponds to the case when the positions of the flipped symbols do not coincide with the positions of the pilot symbols. At low SNR values, the change in the polarity of the pilot symbols may happen either due to the attack or due to the additive noise on each symbol. Therefore, to compute the probability of misdetection, we need to consider the event when the ambient noise unflips the pilot symbols already flipped by Eve. At arbitrary SNR values, the probability of misdetection for a given frame can be computed as

$$P_{miss}^{(l)} = \sum_{j=1}^{L} p^j (1-p)^{L-j} \left[\sum_{x=0}^{j} \binom{L_p}{x} \binom{L-L_p}{j-x} q_l^x \right], \quad (8)$$

where $\binom{n}{r}$ denotes "n choose r" operation, and $q_l = \text{prob}(y_{k,l} < 0 \mid x_k = 1, h_l)$, which is identical to $\text{prob}(y_{k,l} > 0 \mid x_k = -1, h_l)$. To compute the expression in (8), we assume that Eve flips the bits based on tossing a coin independently L times. For a given frame, since the fading coefficient h_l is constant, the value of q_l is determined by the channel realization as well as the additive noise variance.

Similar to the events causing misdetection, events causing false alarm occur when at least one of the received symbols on the pilot locations is flipped due to the additive noise in the case of no attack. The corresponding probability of false alarm can be computed as

$$P_{false}^{(l)} = 1 - (1 - q_l)^{L_p}. \quad (9)$$

For a given SNR value, i.e., for a given q_l, the expression in (9) indicates that $P_{false}^{(l)}$ increases as L_p increases. However, the behaviour of $P_{miss}^{(l)}$, given in (8), as a function of q is not straightforward. As a result, in the rest of this section, we plot $P_{miss}^{(l)}$ and $P_{false}^{(l)}$ against different values of L_p, L and q_l. In Fig. 4, we present the above values when $L = 100$ and when L_p takes values from $\{1, 2, \ldots, 20\}$. The plots in Fig. 4 show that for a given value of q_l, $P_{miss}^{(l)}$ decreases as L_p increases, while $P_{false}^{(l)}$ increases with L_p. However, when q is sufficiently small (i.e., high SNR values), $P_{miss}^{(l)}$ can be reduced while keeping $P_{false}^{(l)}$ within acceptable range. This discussion shows that Bob can accurately detect the presence of Eve at high SNR values by observing the polarity of the received symbols on the pilot locations. Furthermore, with correct detection, Bob can decide whether to combine the received symbols on a secondary tap with the main tap or not. As shown in Fig. 2 and Fig. 3, Bob can be conservative to drop the attacked taps, and just decode the information from the main tap only. In such a case, although there is error performance loss, the receiver can still decode the information bits. On the other hand, if the attack-ignorant receiver combines all the taps without validating the polarity of the pilot symbols, then it would result in substantially degraded error-performance.

In the next section, we consider the case when Eve does not have perfect knowledge of Alice-Bob's channel. With inaccurate estimate of the channel, we explore whether Bob can take advantage of this situation to improve the error performance than that of just decoding from the main tap.

IV. Flipping Attack: Incorrect Channel Estimates

In practice, the knowledge of the channel estimate at Eve may not be accurate. Let the channel estimate of the l-th tap of Alice-Bob's channel at Eve be denoted by $\hat{h}_l = h_l + \epsilon_l$, where h_l is the corresponding estimate at Bob and ϵ_l is the

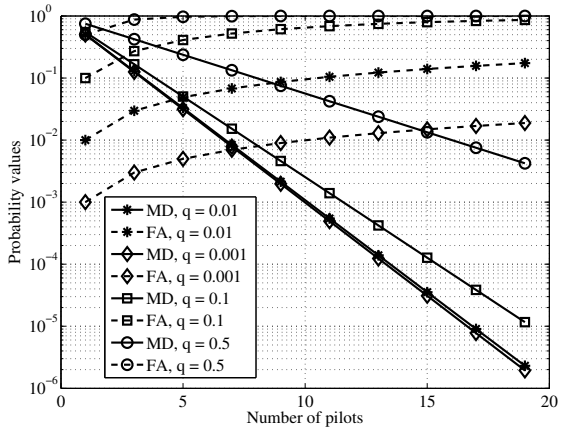

Fig. 4. Probability of misdetection (MD) and probability of false alarm (FA) for various values of L_p and q (we discard the subscript l) with $L = 100$. The receiver detects the flipping attack if at least one of the received symbols on the pilot locations has changed its polarity.

estimation error at Eve. Since we are focusing on one of the secondary taps, we henceforth drop the subscript l in this section. With flipping attack using incorrect channel estimate at Eve, the received symbol at Bob is of the form

$$y_k = (-h - 2\epsilon)x_k + z_k.$$

In the above expression, Eve has attempted to flip the transmitted symbol from x_k to $-x_k$. However, because of incorrect channel estimate, the received symbol at Bob is offset by $-2\epsilon x_k$ compared to the case of perfect estimate.

When the estimation error ϵ is non-negligible, we will show that Bob can identify the flipped data symbols, and subsequently undo the modifications to some extent. Let us assume a frame based communication between Alice and Bob with L denoting the length of the frame and L_p denoting the number of pilot symbols, which are transmitted at random positions in the frame. Since the positions of the pilots are generated based on a shared key, Eve does not know the pilot locations. Furthermore, since the pilots are also BPSK symbols, Bob can distinguish between a flipped and unflipped pilot symbol by observing the polarity of the received symbols. For the sake of exposition, we use an indicator variable $\mathcal{A}$ to represent the attack event. Within a frame, the expected value of the received symbols corresponding to the flipped BPSK symbols are

$$E\{y_k|x_k = 1, \mathcal{A} = 1\} = -h - 2\epsilon, \tag{10}$$

$$E\{y_k|x_k = -1, \mathcal{A} = 1\} = h + 2\epsilon, \tag{11}$$

where the expectation $E\{\cdot\}$ is over the symbols of the frame. In (10) and (11), $\mathcal{A} = 1$ indicates the attack event. In the case of no attack, similar values are given by

$$E\{y_k|x_k = 1, \mathcal{A} = 0\} = -h, \tag{12}$$

$$E\{y_k|x_k = -1, \mathcal{A} = 0\} = h. \tag{13}$$

If L_p is sufficiently large, Bob can obtain the estimates of the above statistics in (10)-(13) by observing the pilot symbols.[2]

[2]In order to obtain the statistics in (10)-(13), we assume that Eve has flipped some pilot symbols from -1 to 1, and some from 1 to -1.

Then, if the difference $|E\{y_k|x_k = 1, \mathcal{A} = 1\} - E\{y_k|x_k = 1, \mathcal{A} = 0\}|$ is greater than 2σ, then Bob proceeds to undo the flipping attack as discussed below.

Using the above estimates, Bob will observe the rest of the $L - L_p$ symbols (the data symbols) in the frame, and then identifies the symbols that were flipped by Eve. The rationale behind this approach is that the flipped symbols are likely to be closer to $-h-2\epsilon$ or $h+2\epsilon$, instead of $-h$ or h. Using this idea, Bob first identifies the locations of the data symbols that are closer to $-h-2\epsilon$ or $h+2\epsilon$ than $-h$ or h. Let these locations be denoted by $\mathcal{I} \subset \{1, 2, \ldots, L\}$. Similarly, Bob identifies the locations of the data symbols that are closer to $-h$ or h than $-h-2\epsilon$ or $h+2\epsilon$. Let these locations be denoted by $\bar{\mathcal{I}}$. With this information, Bob changes the polarity of the received symbols on $\mathcal{I}$, while retaining the polarity of the symbols on $\bar{\mathcal{I}}$. Finally, the updated received symbols $\{y_k\}$ are suitably combined with those of the main tap in order to decode the information symbols. Specifically, for each symbol y_k, Bob combines it with the main tap depending on his confidence on how close the received symbol is to the offset versions. In particular, Bob generates an LLR as follows

$$\Delta_k = \log\left(\frac{e^{\frac{-|y_k-(h+2\epsilon)|^2}{\sigma^2}} + e^{\frac{-|y_k-(-h-2\epsilon)|^2}{\sigma^2}}}{e^{\frac{-|y_k-h|^2}{\sigma^2}} + e^{\frac{-|y_k+h|^2}{\sigma^2}}}\right), \tag{14}$$

which quantifies his confidence on whether the received symbol is close to $\{-h-2\epsilon, h+2\epsilon\}$ or $\{-h, h\}$. Here, σ^2 is the variance of the additive noise. From the above confidence metric, when $\Delta_k < -\delta_{th}$, for some optimization parameter δ_{th}, Bob first changes the polarity of y_k before including it with the corresponding symbol from the main tap. On the other hand, when $\Delta_k > \delta_{th}$, Bob uses y_k as it is before including it with the corresponding symbol from the main tap. Finally and importantly, when $-\delta_{th} \leq \Delta_k \leq \delta_{th}$, Bob discards y_k, which implies that his confidence is not high to decide whether the symbol is flipped or otherwise. It is intuitive that when ϵ is small, Bob cannot confidently distinguish between flipped and unflipped data symbols, and therefore, the best strategy is to neglect those received symbols on the tap. Otherwise, combining the symbols despite low confidence would only degrade the error performance as explained in the previous section.

A. Simulation Results

In this section, we present simulation results to demonstrate that Bob can leverage on the estimation error at Eve to improve his error performance compared to the conservative option of just using the main tap. The simulation setup in this section is same as in Section III. However, a major distinction is that the knowledge of the channel estimate at Eve is not accurate. We have chosen the estimation error to be either 50% or 100% to drive the point that Bob can significantly improve the performance. Bob first computes the estimates of the expressions in (10)-(13) using the pilot symbols, and then classifies the data symbols as either flipped or otherwise. The coded BER performance with incorrect estimate of the two-tap channel is presented in Fig. 5. In the presented results, "A-2, C-1" denotes the case when the second tap is attacked,

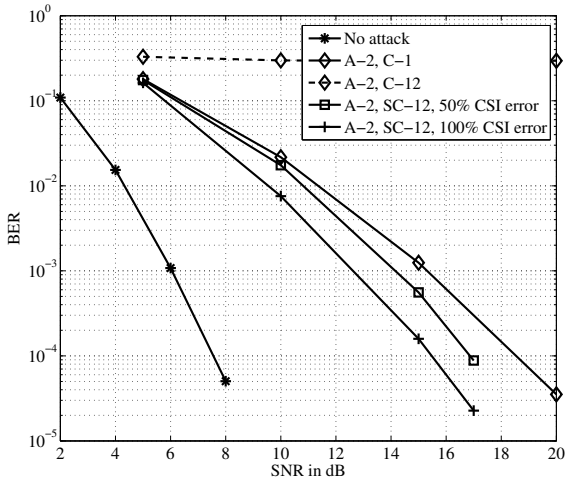

Fig. 5. Turbo-coded error performance on a two-tap channel with imperfect estimate of Alice-Bob's channels at Eve. With channel estimation error at Eve, Bob opportunistically distinguishes between flipped and unflipped data symbols to some extent to improve the error performance.

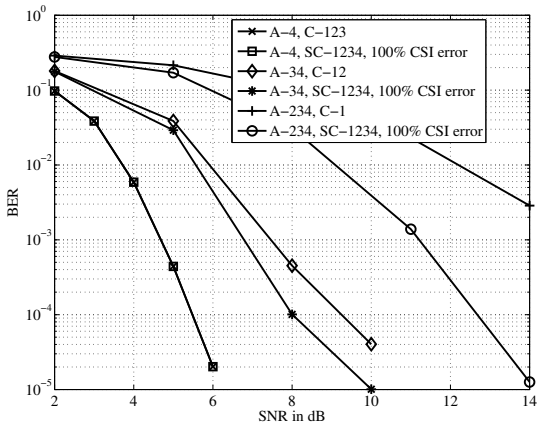

Fig. 6. Turbo-coded error performance on a four-tap channel with imperfect estimate of Alice-Bob's channels at Eve.

whereas Bob uses first tap to obtain the LLRs. Similarly, "A-2, SC-12" denotes that second tap is attacked, and Bob smartly combines the symbols on the second tap with that of the first tap based on the LLR in (14). The plots in Fig. 5 show that with larger values of ϵ, Bob can opportunistically use the secondary tap to his advantage. We have used $\delta_{th} = 0.5$ as the threshold to distinguish the flipped and unflipped data symbols.

We conducted similar experiments on the 4-tap channel with average power-profile $\{0.4, 0.3, 0.2, 0.1\}$, and the corresponding results are presented in Fig. 6. We have assumed 100% channel estimation error at Eve in this case. To obtain the results we used $\delta_{th} = 1$ for all the cases. Similar to the plots in Fig. 5, the plots in Fig. 6 also show that the estimation error at Eve can be opportunistically used to improve the error performance at Bob. It is interesting to note that the BER improvements from unflipping the symbols on the fourth tap (in the case of "A-4, SC-1234") is negligible, whereas BER gains from unflipping the received symbols on the second, third and the fourth taps (in the case of "A-234, SC-1234") are significant. This behaviour is attributed to the fact that the fourth tap alone contributes negligible signal power, whereas the signal power contributed by the second, the third and the fourth taps together are comparable to that of the main tap. We highlight that the choice of δ_{th} is crucial in reaping BER improvements from the attacked taps. While smaller values of δ_{th} include symbols on the unreliable taps into the decoding process thereby degrading the performance, larger values of δ_{th} forces Bob to discard the received symbols on the attacked taps, thereby matching the performance of that of combining the unattacked taps.

V. Discussion and Directions for Future Work

We have discussed a strong adversarial attack on DSSS systems wherein the attacker instantaneously modifies the transmitted symbols such that some of the delayed copies carry no information on the transmitted data. Unlike the jamming attack, this attack when perfectly executed, cannot be detected at Bob by measuring the SINR variations. Perfect execution of this attack necessitates the attacker to accurately know all the channel realizations in the model. Given that DSSS uses wideband communication, all the underlying channels in the model are frequency selective, and this implies that Bob may also receive multiple copies of the manipulated symbols transmitted from Eve. In this work, we have assumed that Eve nulls all the multipath components that she generates by converting the Eve-Bob's channel from frequency-selective to frequency-flat. However, in practice, this assumption needs unlimited power, and as a result, Bob may also receive multiple copies of the manipulated symbols from the attacker. It is interesting that Bob, who is oblivious to the presence of the attacker, may see more taps than that in Alice-Bob's channel, and the total number of taps depends on whether the delay profiles of Alice-Bob's and Eve-Bob's channels coincide. How can Bob opportunistically take advantage of no or imperfect nulling of multipaths in Eve-Bob's channel is an interesting direction for future work.

References

[1] T. H. Morris et al., "Engineering future cyber-physical energy systems: Challenges, research needs, and roadmap," *41st North American Power Symposium*, Starkville, MS, USA, 2009, pp. 1–6.

[2] George Loukas, *Cyber-Physical Attacks: A Growing Invisible Threat*, Elsevier Inc. 2015

[3] M. Duarte and A. Sabharwal, "Full-duplex wireless communications using off-the-shelf radios: feasibility and first results," *Forty Fourth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2010, pp. 1558–1562.

[4] Z. K. M. Ho and E. A. Jorswieck, "Instantaneous Relaying: Optimal Strategies and Interference Neutralization," in *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6655–6668, Dec. 2012.

[5] Q. Wang, Y. Dong, J. Zhao, N. Li, J. Qian and B. Liu, "Instantaneous Relaying: Feasibility Conditions for Interference Neutralization," in *IEEE Communications Letters*, vol. 19, no. 8, pp. 1370–1373, Aug. 2015.

[6] A. El Gamal and N. Hassanpour, "Relay-without-delay," in the proc. of *IEEE ISIT 2005*, Adelaide, SA, 2005, pp. 1078–1080.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 2006

[8] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang. "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE," *IEEE Journal of Solid-State Circuits*, Vol. 46, No. 01, pp. 8-17, 2011.