Research Journal

Darren Lund

July 5, 2018

1 May 2, 2018

Progress: Initial go through, understanding problem and pointing to materials to research/use. Found one paper on arXiv.org that seems to talk about a similar project.

Problems: Measurement of success?

Plans: Beam forming in a random way.

Send a few packets until you receive enough for a key. Receiver sends key from packets received (random selection of packets). Using secrecy code, emitter sends all other packets.

GOAL: Here's a practical way to make SURE that Bob received different packets from Eve.

Papers:

2 May 3, 2018

Progress: Looking at beam forming and trying to understand how it works mathematically. Made a jupyter notebook to try and graph the signal strength/intensity for a single shift, but it doesn't seem to be working.

Problems: Can't get a mathematical formula for a specific signal beam. Trying to utilize info found from Phased Array, Diffracrtion, and Analog Beamforming.

Plans: Find a way to fully understand beam forming. Currently, the thought is to split message into packets, split packets into pieces. If you need k_1 packets to form the message, and k_2 pieces of packet P_i to recreate

P_i, then send k_2 pieces of at most $k_1 - 1$ packets sweeping the beam. If the beam is created with at least one dead zone (and this dead zone sweeps past Bob) then it will probably sweep past Eve as well. If you also change the intensity, then you should be able to get it to just reach Bob sometimes.

Papers: Maybe Intro to Virtual Phased Arrays and Beamforming Applications and/or Antenna Arrays and Beamforming?

3 May 7, 2018

Progress: Trying to understand beamforming, I decided to refresh my understanding of Fourier transforms, which lead me to this video. At the end, it states that the Fourier transform is directly linked to the Heisenberg Uncertainty Principle (HUP), which may be something to explore in the future. Check this video for a bit of intuition on the HUP. Re-evaluated problem, downloaded software to try and assist.

Problems: Can Eve theoretically jam Bob's response? Is there a possibility to ”launch” signals and have them curve?

Is there a mathematical way to represent the ideas of beamforming and jamming?

Plans: Go over a few more of the lecture videos from last year, and start reading Array Signal Processing, found online.

Papers: This seems promising. Array Signal Processing EDIT: this wasn't that helpful. Student project that is ill written and not easily understood.

4 May 8, 2018

Progress: Found something that seems mathematically helpful. See first paper listed in Papers today. Also came up with a few more ideas about how to guarantee that Bob receives something that Eve doesn't.

Problems: Still mostly just the physics of the system. I think that if I can model it and reasonably indicate the signal to noise ratio at any given point, then testing the idea in plans from today and May 3rd should give an

indication of whether or not it will work.
Can you control the range (distance) of a signal? If yes, see second ”big idea”.

Plans: Use jamming as well. Create beams of noise that flank your beam, as well as beams that cross your beam at every possible distance point. This should guarantee that if Bob and Eve are in different locations, then different packets will be jammed at their respective positions.

.

BIG IDEA:
If Alice ”sweeps” her ”shielded” beam around where Bob is in a specific order (not sending enough packets to get the full message), and Bob responds with which packets he received, Alice should be able to use the angles of her beams to determine the arc-length corresponding to where Bob is distance wise from Alice. She can then use this to get the distance to Bob and narrow here ”shield” so that it diverges right before it gets to Bob. This would require Eve to be ”behind” Bob (with respect to Alice) in order to avoid the shield. This, however, would then mean that Eve has a worse channel than Bob, and other methods can be applied.
ASSUMPTION: Bob and Alice are stationary relative to each other.

.

BIG IDEA:
Propagate two signals simultaneous, the desired signal and the negative of it. IF you can determine the range of the signal, then if you have a lower bound for Bob’s distance, Alice can transmit the negative to fade out before it reaches Bob, shortening the ”distance” where the signal is viable (i.e. isn’t canceled out by it’s negative counterpart).
ASSUMPTION: It is possible to control the range of a signal to some degree of percision.

Papers: THIS seems the most helpful so far.

5 May 9, 2018

Progress: I think I have a better idea of how to compute the signal strength. Functional Beamforming (FB) seems to be a good way to implement the ”shield” idea. If you have a beam of noise, containing a much narrower beam of the signal, then you can point them in the same direction and get a narrow beam with high SNR and a ”shield” of practically just noise. Maybe you could improve this further by having another beam (as narrow or narrower than the signal) that cancels out the noise you’re producing in the wider beam?

Problems: What if you only have a short message to transmit? Can this scale down to smaller message sizes? What kind of distance/range are we thinking about? Is reflecting off of something in space practical, or overkill?

Plans: TOMORROW : Make a list of all questions/concerns you currently have to ask on Friday. .
BIG IDEA:
Maybe what you want to do is send a generic beam with the first packet out in Bob’s general direction until Bob transmits that he’s received it. Once that happens, you narrow the signal beam, shift it a bit, and transmit the next packet. If Bob doesn’t signal that he received it after a while, shift the beam to encompass one of the portions included in the previous beam, but excluded in the current one and transmit the SAME packet. Repeat this process using what you can to narrow in on Bob’s exact position. If the noise beam encompasses the entirety of the signal beam, then Eve’s only chance of receiving the message would be if she also was inside the signal beam, i.e. between Bob and Alice.
For the distance part from last time, if you can control when a signal starts to ”fade”, then you want the noise to fade almost completely at Bob, and the actual signal to begin to fade just past Bob.
ASSUMPTION : We know the distance to/position of Bob’s device.

Papers: MATH! **THIS IS THE IMPORTANT PAPER SO FAR**.

6 May 10, 2018

Progress: Better understanding of how the channels work. I’ve also started creating a class in Python that seems like it will be helpful for mimicking a lot of MatLab’s stuff (at a lower level). This is important/necessary because I don’t know how to test my ideas in MatLab, so trying to figure out how their tools work, I can then create one that is specific to the implementation I want.
Well, I tried it in python and while I definitely need it to understand

things better, things are obviously faster in MatLab...so the next step is to try and figure out how to make the model to test ideas in MatLab.

Problems: I still don't know how exactly these channels work. I think what I want to do is make a plot of probable bit error rate as a function of position and beams being used.

Plans: See about graphing bit error rate as a function of position and beams being used. Then I can get a better feeling for how changing the beams will increase/decrease the probability that Eve receives the message as well.
Read a LOT of MatLab documentation.

Papers: I think this BCC paper should be helpful. If nothing more, it helps me see a mathematically heavy project in this field.

7 May 11, 2018

Progress: Coded up the function for the generalized functional beamformer. Learned a bit more about implementing things in MatLab/Simulink from Zach, as well as physical nature of communications from Marco.

Problems: I think my shielding idea won't work. based on a discussion Marco had with Zach about getting an approximation for ambient noise to filter it out should mean that as long as Eve can listen to the part without the signal simultaneously, she can filter the added noise.

Plans: Study the effects of different power levels and how to introduce fading into the system.

Papers:

8 Questions for Meeting: May 14, 2018

1. Which basic channel model should be implemented? Is it the basic wiretap channel, does it include position relative to the transmitter/receiver, ..?

2. Is this supposed to be a theoretical, mathematically based proof, or an empirically tested one?

   (a) If it's theoretical, how can I write the probability that Eve receives a packet based on the beam(s) I'm generating/signals present at her position?

   (b) If it's empirical, which models should I use? Do you already have the algorithms for the channel coded up/are they easily emulated?

3. Project Clarification

   (a) What is the final "end" goal quantitatively?

   (b) How can I measure the success of my ideas?

   (c) What point do you want me to get to so your algorithms can be utilized?

4. How computationally heavy IS beamforming? How is it used in practice?

5. What does the Cross Spectral Matrix do in beamforming? Physically, what does it mean/how is it used?

9 May 15, 2018

Progress: It seems that the best option for implementing the channel is to use some of the stuff from the slides João provided to try and model fading in it's 3 primary forms.

Want to find the cardinality of the set of packets that Bob received that Eve didn't, and show that this expected cardinality is greater than or equal to 1. Let N be the set of all packets being transmitted, B_l the event that Bob receives packet l, and E_l the event that Eve receives packet l, then I want $|\{l|B_l \wedge \neg E_l\}_{l \in N}| \geq 1$.

Problems: I need to have some kind of probability to describe the successful reception of a packet or not. The paper that I found today seems promising, and contains a calculation for that in a MISO (Multiple Input Single Output) model that uses delated feedback, but it's dense. It will take some time to figure out what everything means, and likewise, how I can adapt it to fit the channel model I'll eventually have.

Plans: Keep going over all the specifics for fading in a wireless channel so I can build a good model. I also might want to look up what different models have already been coded up through File Exchange, as suggested by Marco.

Papers: Packet Error Probability of a Transmit Beamforming System with Imperfect Feedback.

10 May 16, 2018

Progress: I believe I want to use a Rayleigh Fading model. Everything I've looked at has implied either Rayleigh or Rician, but Rician is only used when there is a clear line of sight, which usually isn't the case. Rayleigh fading is normally distributed complex i.i.d, with $\mu = 0$ and $\sigma^2 = \frac{1}{2}$ for both real and imaginary components. I.E., x and y are normally distributed with parameters described above, and $z = x + iy$. This gives $a(t) = \sqrt{x^2(t) + y^2(t)} \Rightarrow \mathbb{E}[a^2(t)] = 1$ which implies the signal-to-noise ratio is not affected by fading.

You want to maximize the probability that Bob receives a packet and Eve doesn't (which is equivalent to maximizing the probability that Eve receives a packet that Bob doesn't). If inside the beam there's a certain probability that any given receiver gets a packet (and this is the same regardless of where in the beam you are) then this means that

$P(B_l) = P(E_l)$, and $P(\neg B_l) = P(\neg E_l)$, giving us $P(B_l) + P(\neg E_l) = 1$, as well as $P(B_l \wedge \neg E_l) = P(B_l)P(\neg E_l) = C$ where our goal is to maximize C. So, letting $P(B_l) = x$ and $P(\neg E_l) = y$, we have maxxy subject to $x + y = 1$, which has solution $x = y = \frac{1}{2}$.

Problems: I'm still not sure which assumptions are good and what channel models to use. So far I have: Assumption 1 - Bob and Eve are both always in the beam (because if either of them isn't, then the probability of them receiving the packet is 0, so that case is trivial; Assumption 2 - The channel is constant over the time it takes to transmit a single {bit/symbol/packet}.

Plans: Use formulas I have to think about how best to model the channel. Include fading, path loss, shadowing, gaussian noise...?
Figure out how to manipulate the probability of receiving a packet inside the beam.

Papers: Rayleigh Fading stuff.

11 May 17, 2018

Progress: Wrote down what the problem looks like mathematically (more or less) and am working on a way to make it solvable.

Problems: Right now, I need a way to relate the probability that Bob gets an erroneous bit and the probability that Eve gets an erroneous bit. Otherwise, the system is underdetermined and I don't really know where to go from there with it.

Plans: Currently I'm running a simulation to approximate BER at certain distances/angles for a specific beam and then use those values in order to calculate the probability of receiving a packet at that position. Then use that "data" to approximate the following: if Bob is located at (r, θ), then if Eve is located at {position in graph}, the value is the probability that Bob receives a packet that Eve didn't (under some setup of packet length and whatnot).

Papers: Nakagami Fading seems to be a more general/flexible model of both Rayleigh and Rician fading. This paper lays out a lot of the thought behind calculating the probability of packet error, and should be kept.

12 Meeting: May 21, 2018

Progress: Created a few different algorithms for approximating BER with the beamforming ideas, but I'm not sure how valid they are. I've looked into a bit of the math behind beamforming and channel modeling, and, given a few assumptions, have made some progress in determining how to ensure Bob gets something Eve doesn't. In essence, the thought is to maximize the probability that Bob got a packet that Eve didn't. However, if Bob's and Eve's probabilities are independent, and simply a function of the channel, then they should be equal. Thus, if B_l is the probability of Bob receiving packet l, and E_l the probability of Eve receiving it, then we want to maximize $B_l(1 - E_l)$ subject to $B_l = E_l$ which occurs when these are both $\frac{1}{2}$.

Problems:

1. I'm not sure how valid my channel modeling is; especially concerned with how to translate the linear algebra I'm using into meaningful statements.
2. I don't know if there's a way to hinder Eve's channel without also hindering Bob's; likewise, I don't know if there's a way to improve Bob's channel without improving Eve's.
3. How to represent the difference between Eve and Bob's channel so that I can represent who got what?
4. What exactly does the covariance/variance represent in the case of communications?

Plans:

1. Working through the math more to see what comes out.
2. Verify my algorithms/find ones that make more sense.
3. Try to simulate actual situation.

Papers:

1. Nakagami Fading
2. Packet Error Probability of a Transmit Beamforming System with Imperfect Feedback
3. Packet Error Rate in the Non-Interleaved Rayleigh Channel

13 Meeting: May 25, 2018

Progress: Looked more into Beamforming and found some papers about it and what people are using it for. Mostly it's just a nice tool to increase the signal power in a specific direction, while simultaneously decreasing the signal power in others. People use different methods and ideas to decrease the side lobes, though for all of them, they don't connect the math to the physical representation of the signals. I understand better why the math comes out the way it does, but I don't have it all down yet.

I also believe I know the assumptions that I want to use for the channel:

1. MISO situation where Alice has $L \geq 2$ antennas
2. The channel experiences AWGN and Rayleigh fading (distance attenuation?)
3. Probability of receiving a package is based on SINR as a function of angle (and distance?) from Alice (using Cyclic Redundancy Check to verify packets).

Problems:

1. Should I be attacking SINR, or actually evaluating probabilities of packet error?
2. I'm not quite sure what my next step should be, or where I should be taking things.

Plans:

1. Packet Loss Probability as function of SINR.
2. SINR as a function of θ/beam.
3. Packet loss comparison between Bob and Eve as a function of angle between them (relative to Alice).

In general,

1. Dig more in depth in papers that I think are helpful.
2. Reframe problem so that I understand better where I'm at and what I'm working towards.

Papers: See Papers folder.

14 Meeting: June 1, 2018

Progress: Coded up calculations for SNR, BER, and PLP (Packet Loss Percentage). Current example in code has the following properties:

1. Transmitters : 7
2. Wavelength of : 1
3. AWGN variance : 1
4. Rayleigh variance : 1
5. Distance Attenuation Exponent : 2
6. Packet length : 10 bits
7. Number of Packets sent : 51
8. BPSK modulation with CRC

Problems:

1. Distance/SINR units.
2. Complex values.
3. Optimal PLP.
4. Need to go back and list assumptions made while writing code for checking.
5. Someone else looking over the equations in use would be nice.

Plans:

1. Go deeper into equations to understand them better.
2. Look at PLP as a function: what variables go into it and how do those variables effect it.
3. Go back to Functional Beamforming and see what's up.
4. Consolidate code to make it more accessible.

Papers: BER from SINR from different modulation schemes.

15 Meeting: June 8, 2018

Progress:

1. Tried to figure out the physics of transmitting for Functional Beamforming, but didn't make much progress.
2. MISO should easily extend to a MIMO setting, so we can focus on just MISO without loss of generality.

Problems:

1. Nothing I've read seems to help with using reciprocity to implement Functional Beamforming in the transmit mode.
2. Model is currently spread out. Need it condensed for analysis.

Plans:

1. Write up my model: assumptions made, equations used, where those equations come from and why they're valid, etc.
2. Read the referenced papers.
3. Expand on code to simulate sending a signal, tracking what occurs. Current thought is to do a colormap-like setup where each point is calculated as whether or not Bob's position received something that point didn't.
4. Go through CITI training (at least half).

First situation: beamform and oscillate every time regardless, and see how it works

Papers: See Papers folder in Google Drive.

16 Meeting: June 19, 2018

Progress:

1. Typed up most of the bare bones for calculations thus far.
2. Read one of the two referenced papers.
3. Finished CITI training.

Problems:

1. Not sure whether I should simulate the entire process of generating a signal, transmitting, and then receiving, or should just use the calculated Packet Loss Probability to arbitrarily drop packets.
2. Don't have access to Stopping Sets for Physical-Layer Security paper.

Plans:

1. Continue to write up my model: organize thoughts/sections, fill out ideas, etc.
2. Continue to read and understand the referenced papers.
3. Expand on code to generate needed/desired figures. Current thought is to do a colormap-like setup where each point is calculated as whether or not Bob's position received something that point didn't.

Papers: See Papers folder in Google Drive.

17 Meeting: June 28, 2018

Progress:

1. Built the ”easier” simulator.
2. IT SEEMS TO WORK!! =D (at least in this simulator).
3. Ran a few tests.

Problems:

1. Can’t control SINR like I thought. It depends on the random fading which can’t be determined in advance.
2. Fewer valid beam angles than I thought.
3. Simulations take a LONG time. I calculated that to do 1,000 trials of ”sending” 1,000 packets of 51 bits, it would take 9-10 days.
4. Last trial took 32 iterations of sending the packets and >14 minutes.

Plans:

1. Write full mode/speed up short one.
2. Continue to read and understand the referenced papers, and those whose equations are used.
3. Write more.

Papers: See Papers folder in Google Drive.

18 Meeting: July 5, 2018

Progress:

1. Tried to optimizer/reorganize the code (reduced single iteration from 14 min to 1 min).
2. Found a few bugs that I needed to fix (angle and beam orientation).
3. Began a trial (crashed w/out returning anything first time).
4. Each trial is ”sending” 100 packets of 51 bits each.

Problems:

1. I couldn’t really improve the code for now, just reduced the intensity of the trial.
2. I haven’t taken into account distance, but I might need to in order for things to make sense.
3. Don’t have results yet, but with time, that should be fixed.

Plans:

1. Include distance.
2. Find better way to store results.
3. Run LOTS of trials; rewrite code to save it and keep track of it better.
4. Write things up.
5. Make everything a function of Packet Error Probability.
6. COMMENT THE CODE!!

Papers: See Papers folder in Google Drive.