*Short introduction (group and table of contents)…*

**SLIDE 3 – INTRODUCTION TO PLA**

The environment in which we have worked is the PLA (Physical Layer Authentication), which constitutes a cutting-edge approach to enhancing security in wireless communication systems.

PLA leverages the unique physical characteristics of the communication channel to authenticate transmitters. Unlike traditional methods that rely on cryptographic techniques, PLA focuses on properties such as the amplitude, phase, and frequency response of the signal.

There are several notable advantages to using PLA. First, it offers enhanced security because it’s extremely challenging for an attacker to mimic the physical properties of the legitimate transmitter’s signal. Second, it has a low overhead since it doesn’t require the computational resources needed for complex cryptographic algorithms. Lastly, PLA enables real-time authentication, allowing for a quick verification process that is crucial in many applications.

It is possible to count several related works in the PLA environment, a lot of them regarding communications between Bluetooth Low Energy devices. Other related works propose surveys that try to collect different works done in this field, useful to introduce people like us to the PLA environment. Even if there are several related works, none of them propose an effective PLA scheme to authenticate wireless communications, this led us to conduct the experiment we are presenting today.

**SLIDE 4 – OVERVIEW OF THE PROJECT**

…

**SLIDE 5 – EXPERIMENT AND IMPLEMENTATION**

…

(I assume many of these things will be summarized – but put them here anyway)

Now we're going to explain in depth our simulation of how Bluetooth signals behave under different conditions. We created a parametrized simulation environment that allowed us to test various scenarios and fine-tune our Physical Layer Authentication, or PLA, scheme.

Our implementation involved several key steps. First, we combined data and authentication signals. We represented these as binary waveforms with different power levels, using the peaks to distinguish between them. Specifically, the authentication signal was designed with lower power peaks, allowing for more accurate analysis of both components. This approach gave us a unique 'fingerprint' for each transmission.

To make our simulation as realistic as possible, we varied two critical parameters. We tested distances ranging from 1 to 50 meters, covering the typical range of Bluetooth communications. We also adjusted the Signal-to-Noise Ratio, or SNR, from 10 to 30 decibels. This helped us model how signal quality degrades over distance and in different noise environments.

The core of our simulation revolves around a sender transmitting a signal consisting of a key for authentication and a data message, mixed with known power parameters. Within the receiver, we implemented two decoding methods: fixed-threshold and variable-threshold.

For the variable threshold method, we considered four peaks on the signal: high-high (maximum value), low-low (minimum value), and two intermediate values (medium-low and medium-high). These measurements are adjusted based on a 'center' parameter, allowing us to precisely define and update these peak values. This approach helps refine the intermediate values by considering both previous and current measurements.

The fixed threshold decoding is based on the center parameter. We defined a method of bit concordance/discordance to ensure the received signals consistently reflect the originally mixed waveforms.

To determine the authenticity of a decoded message, we set a maximum number of permitted error bits on the key signal. Using Hamming distance, we calculated the number of incorrect bits compared to the original signals. This is crucial for computing the Bit Error Rate (BER), which we use to calculate false alarm and missed detection rates.

Speaking of noise, we didn't forget about the inherent messiness of real-world wireless channels. We applied white noise to our signals, specifically what's known as Additive White Gaussian Noise, or AWGN. This simulated the kind of interference and distortion you'd expect in a real Bluetooth transmission.

Our simulation environment allows for iterative refinement. After multiple executions, we can adjust the tolerance threshold to verify transmission correctness, tailoring it to desired security performance levels. By adjusting all these parameters, we created a flexible, realistic simulation environment. This setup allowed us to design, test, and refine our PLA schemes, giving us valuable insights into how they might perform in real-world Bluetooth communications.