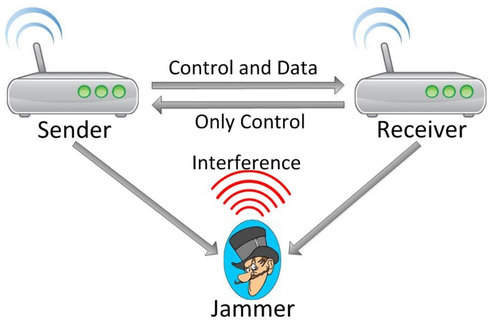
Constrained Adaptive Jamming and Anti-Jamming

Carnegie Mellon University
CyLab

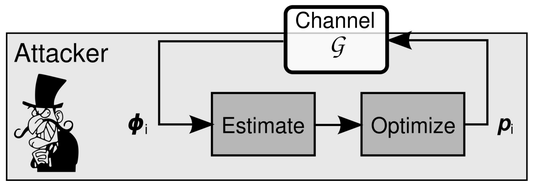
Self-Tuned, Inference-Based, Real-Time Jamming

Bruce DeBruhl, Yu Seung Kim, Zack Weinberg and Patrick Tague

Problem Overview



Goal: To understand the jamming threat we aim to extend the jamming attack space by exploring how low energy, hard to detect, and high impact a jammer can be mounted if an attacker listens and attacks. To do this we introduce Self-Tuned, Inference-based, Real-time jamming or STIR-jamming



STIR-jamming remodels the communication system as a control system with a discrete plant G which maps jamming parameters to observed performance. In this model the attacker:

- Observes the communication system
- Refine a channel model based on observation
- Optimize its attack with the refined channel model
- Repeat

Attack Formulation

mSTIR-Jamming Attack

Given: $\phi_k, \mathcal{S}, \widehat{\mathcal{G}}_k, \boldsymbol{p}_k$

Find error: $\epsilon_k = \text{error}(\phi_k, \widehat{\mathcal{G}}_k, \boldsymbol{p}_k)$

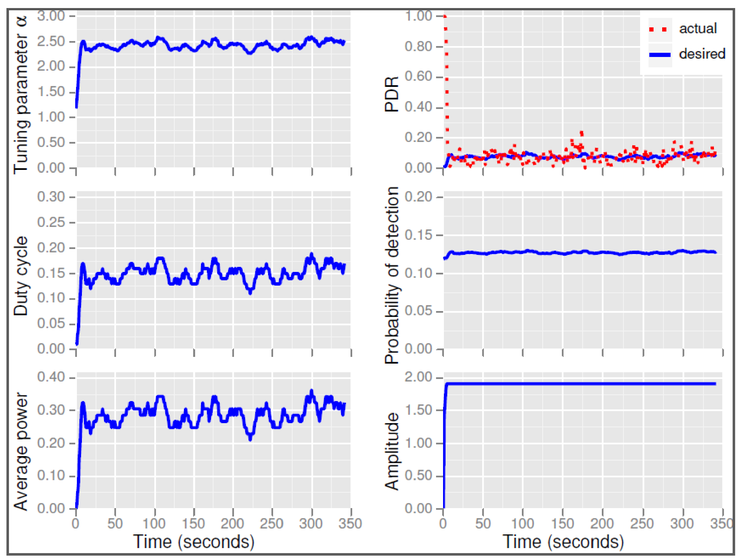
Estimate: $\widehat{\mathcal{G}}_{k+1} = \text{update}(\widehat{\mathcal{G}}_k, \epsilon_k)$

Optimize: $\boldsymbol{p}_{k+1} = \arg\max_{\boldsymbol{p}} \ \mu(\mathcal{S}, \boldsymbol{p}, \widehat{\mathcal{G}}_{k+1})$ s.t. $\boldsymbol{p}_{min} \leq \boldsymbol{p} \leq \boldsymbol{p}_{max}$

We implement a model-based version or the mSTIR-jamming algorithm as below:

- Given a model $\widehat{\mathcal{G}}_k$ and jamming parameter estimates p_k find the error ϵ_k from the observations Φ_k
- Update your model $\widehat{\mathcal{G}}_{k+1}$ given your error ϵ_k
- Optimize the jamming parameter p_{k+1} given $\widehat{\mathcal{G}}_{k+1}$
- Repeat

Implementation Results



- Initial attack implemented with random jamming on 802.15.4 architecture
- Implemented in SDR (USRP2)
- System observations given by the receiver to the jammer with a 10% error
- Measured Actual PDR
- Calculated
  - Average Power
  - Jamming Duty Cycle
  - Jamming Amplitude
  - Estimated Probability of Detection
- Tuning Parameter quickly stabalizes

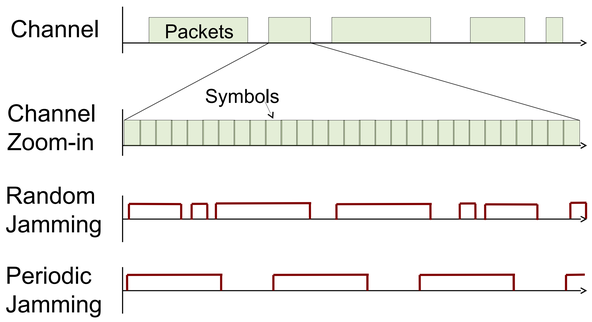
This research was supported by NSF CAREER award CNS-1149582, CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office, the Northrop Grumman Cybersecurity Research Consortium, and an NDSEG fellowship. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, Northrop Grumman, NSF, or the U.S. Government or any of its agencies.

Adaptive Filtering Selection for Efficient Anti-Jamming

Bruce DeBruhl and Patrick Tague

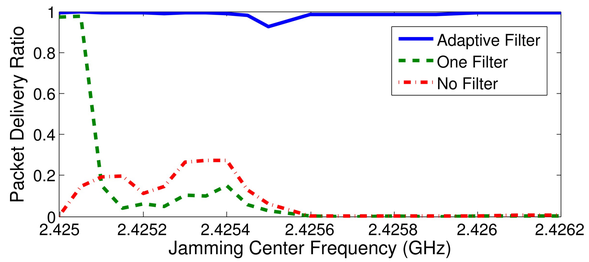
Problem Overview

•*Short Form Periodic Jammers* (SFPJ) aim to deny a wireless channel by alternating between emitting noise and sleeping at speeds on the magnitude of symbol time



SFPJ can *reduce attack cost by over 90%* compared to constant jamming, making it a highly effective attack

- We have shown [DeBruhl and Tague ICCCN'11] that filtering can eliminate short form periodic jamming
- We have also shown [DeBruhl and Tague PECCS'12] that a single filter is insufficient



Problem: To eliminate the effect of SFPJ with a single filter, the center frequency of the jammer must be known

Goal

Develop a technique which mitigates the effects of short form periodic jamming while making no assumptions about its center frequency

Design Strategies

Ideally, our technique will:

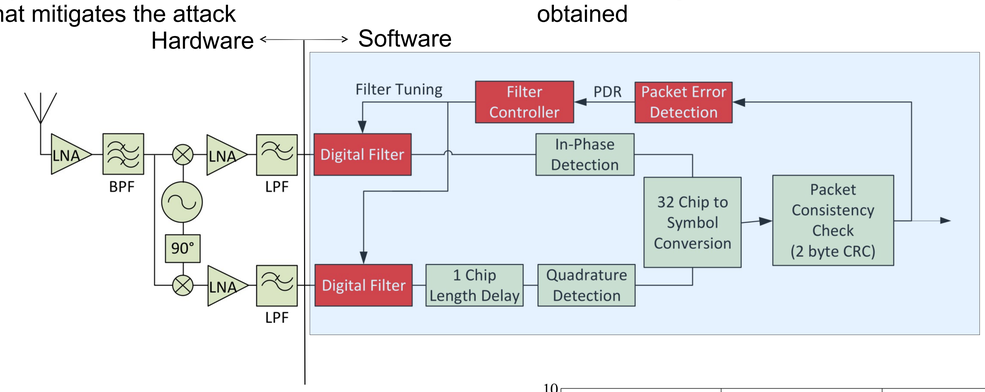
- **Quickly Adapt** to changing jamming strategy
- Not effect performance in benign scenarios
- Be Implemented in **Low Cost Software**

Design & Implementation

- Define a set of filters F such that:
  - Any filter in F is able to be used without decreasing the PDR of the legitimate receiver
  - For any attack frequency fa their must be a filter in F that mitigates the attack

To select a filter the receiver will:

- Use existing detection techniques [Xu, et al. Mobihoc'05] to activate filter search
- Search through filters until a reasonable PDR ratio is obtained



Hand Tuned Filters

- If filter widths are too narrow, it is too hard to find the right filter
- If filter widths are too wide, interferes with DSSS