Eavesdropping on and Emulating MIFARE Ultralight and Classic Cards Using Software-Defined Radio

Ilias Giechaskiel, ilias.giechaskiel@cybersecurity.ox.ac.uk

UNIVERSITY OF OXFORD

CENTRE FOR DOCTORAL TRAINING *in* CYBER SECURITY

Introduction

Contactless protocols are used in ticketing and payment systems but adversaries can decode the radio signals using dedicated embedded devices.

This project aims to eavesdrop on and emulate real transmissions, such as those of access-controlled doors (Figure 1) using Software-Defined Radio.

Figure 1: NFC Door with our Antenna

Contributions

- Open-source Software-Defined Radio NFC demodulator
- Eavesdropping on MIFARE Classic and Ultralight cards
- Software emulation (with encryption) of readers and tags
- Hardware jamming of real reader-tag transmissions

Setup

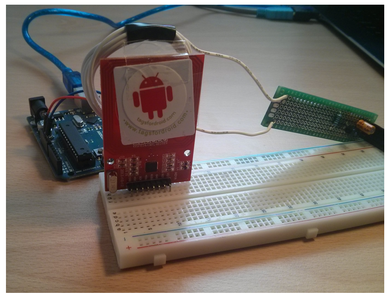

Figure 2: Arduino Module, Tag, and Antenna

The reader is an RFID-RC522 Arduino module, interacting with MIFARE Ultralight and Classic tags. A wire wrapped into a coil acts as our antenna for the USRP N210 SDR. For strength consistency, the antenna and the card are attached to the NFC reader (Figure 2).

Implementation

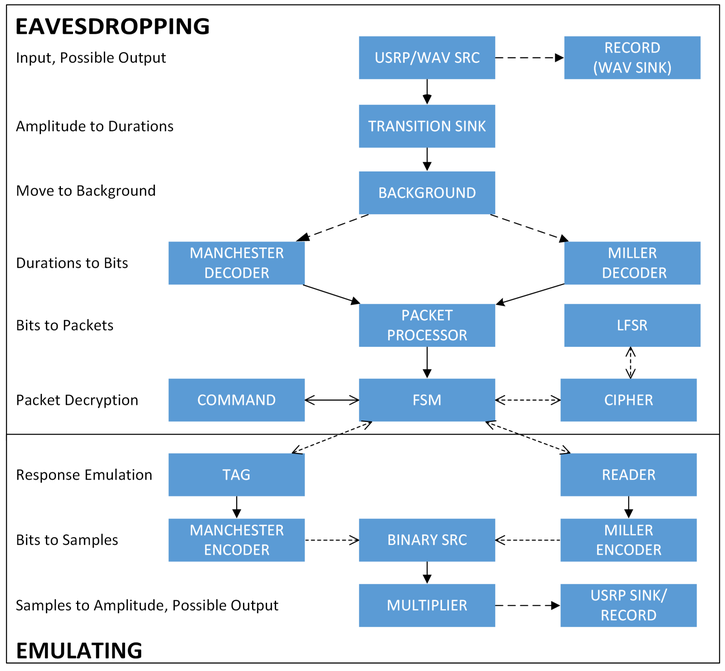

Figure 3: Code Organization: (Dashed) Arrows Represent (Optional) Information Flow

- For demodulation, we keep a running average of the amplitude
- The reader (tag) transmission starts when the signal envelope falls below 10% (rises above 110%) of the average, and uses a Miller (Manchester) encoding
- A state machine interprets bits as commands after decryption
- For emulation, commands are encoded and modulated

Figure 3 illustrates the code interactions, while Figure 4 contains a tag transmission example.

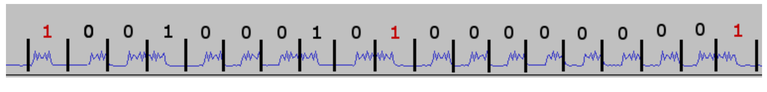

Figure 4: Envelope of an ATQA (0x44 0x00) Tag Command (Start and Parity Bits in Red)

Evaluation

- Prototype achieves interception, decoding, and decryption in 2.2s with off-the-shelf components
- Moving average responds well to sudden signal changes
- Demodulation works even for unknown protocols
- Software emulation is functionally complete. Hardware emulation introduces lags, but is sufficient for jamming
- A more expensive antenna or amplifier can improve range

Conclusions

Prototyping in Python results in good performance-complexity trade-offs for non-real time eavesdropping. Future work could focus on stricter timing requirements, creating a better antenna, implementing newer protocols, and testing real scenarios.

Additional Information

For the report and the code, please visit `https://github.com/giech/usrp_nfc` or scan the QR code to the right. My personal website can be found at `ilias.io`.

Acknowledgements

This project was supervised by Kasper Rasmussen, and external guidance was provided by Simon Crowe. My DPhil is supported by the Clarendon Fund, the EPSRC, Kellogg College, and the CDT in Cyber Security.

Clarendon

EPSRC Engineering and Physical Sciences Research Council