Analyzing Application-Layer Security in Bluetooth Devices: Auditing for Encryption

Madison Tandberg and Dr. Travis Peters
Gianforte School of Computing, Montana State University

Introduction

- Bluetooth is a wireless technology used in a variety of settings, including home, work, transportation, and healthcare. If Bluetooth devices are not properly secured, there can be significant harm to users.
- Data encryption is a widely accepted security control that protects against many weaknesses in data transmission [1].
- Although the Bluetooth protocol establishes encryption standards for packet transmission "over the air", devices are vulnerable to attacks that steal or manipulate data within a device if they lack internal security [2].
- This research aims to develop a technique that utilizes a suite of statistical tests to automatically detect if connected Bluetooth devices are using application-layer encryption.

Background

Bluetooth Device A: Smartphone

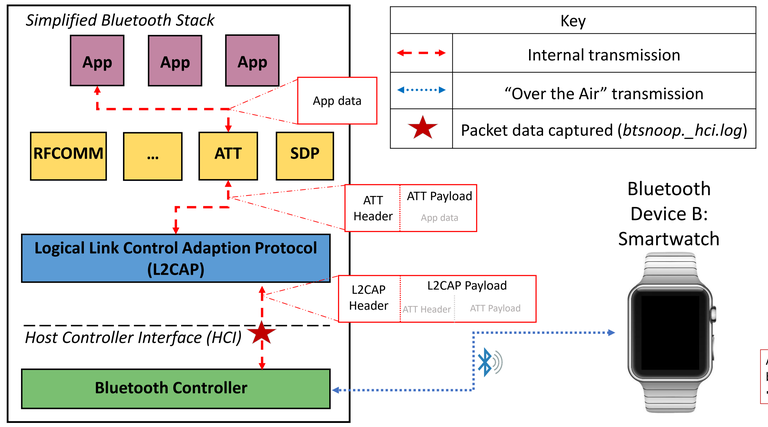


Figure 1: Example Bluetooth network (diagram of Bluetooth device internals adapted from [2])

Bluetooth Networks

- Connected Bluetooth devices exchange data wirelessly through a rapid transmission of packets on the 2.402-2.480 GHz range of radio frequencies.
- Android smartphones provide the ability to collect exchanged Bluetooth data at the Host Controller Interface, which presents an opportunity to analyze the security of internal data transmissions (Figure 1).
- The L2CAP protocol is the primary layer for enabling applications to exchange data between Bluetooth devices, and the Attribute Protocol (ATT) is an additional protocol layer for Bluetooth Low Energy (BLE) devices. Most devices in our dataset are BLE[1].

NIST Statistical Testing Suite (STS)

- The National Institute for Standards and Technology (NIST) provides a Statistical Testing Suite (STS) [3] that quantifies the randomness of a bit stream (0's and 1's of variable length) through a series of statistical tests.
- An encrypted bit stream should be indistinguishable from a random bit stream; all elements of the sequence should have uniform distribution and lack any measurable structure.
- The determination of randomness is based on a *p-value* returned from each test. The *p-value* represents "the probability that a perfect random number generator would have produced a sequence less random than the sequence that was tested" [3].

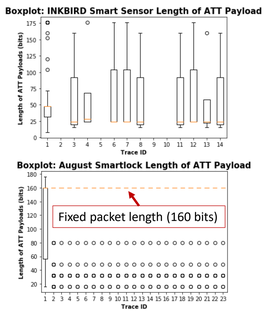
Approach: BLE Attribute Protocol (ATT) Packet Analysis

1. Bluetooth devices communicate.

2. Download and parse *btsnoop_hci.log* to isolate and extract app-layer data.

Packet ID	ATT Payload
1	0001010100011101 0000000000001010 0001111000000000 1101000110100000
...	...
N	0011010100000000 0000101010101101 0011100001110001 0100101001000010

3. Analyze ATT payload (e.g., inspect length, remove app-layer encryption structure).

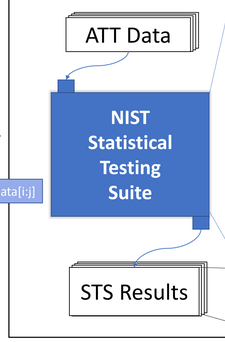


4. Aggregate ATT payload data across multiple packets and/or traces.

ATT Data

```
00010101000111010000000000001010
00001111000000000011010001101000
00100000010101000111010000000000
00010100001111000000000011010001
10100000100000010101000111101000
00000000010100001111000000000011
01000110100000100000010101000111
10100000000000010100001111000000
00001101000110100000100000010100
00011010001110100000000000001010
10000000001101000111010000010000
00101010001110100000000000001010
00001111000000000111010001000111
10000000001101000111010000000000
```

5. Test ATT Data.

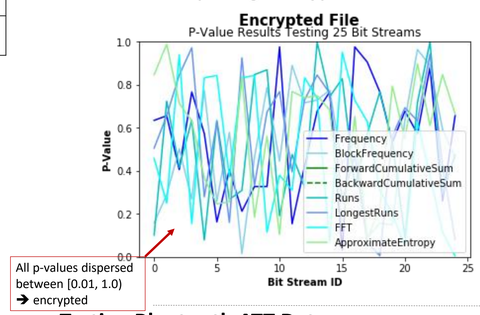


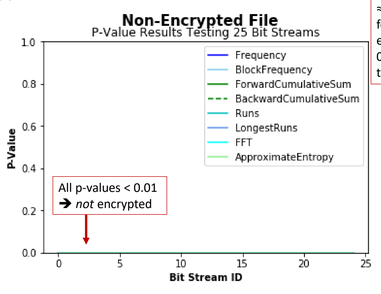

```
NIST_STS(ATT_Data, numBitSeq, ...):
  i = 0
  j = len(ATT_Data)/numBitSeq
  while j <= len(ATT_Data):
    Frequency(ATT_Data[i:j])
    BlockFrequency(ATT_Data[i:j])
    CumulativeSums(ATT_Data[i:j])
    Runs(ATT_Data[i:j])
    LongestRun(ATT_Data[i:j])
    FFT(ATT_Data[i:j])
    ApproxEntropy(ATT_Data[i:j])
    i += j
    j += j
```

Results of STS written to files:
- finalAnalysisResults.txt
- /experiments/{test}/results.txt
- ...

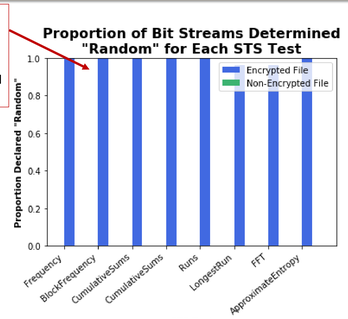
Results of Packet Analysis

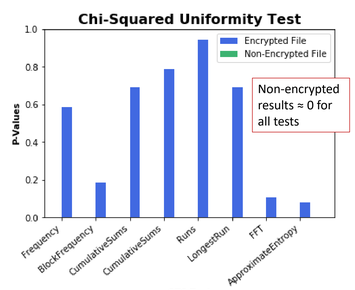
Baseline: Comparing Encrypted[2] and Nonencrypted Data



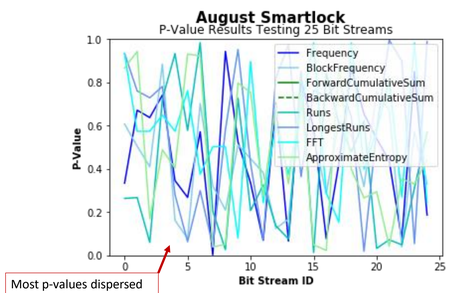


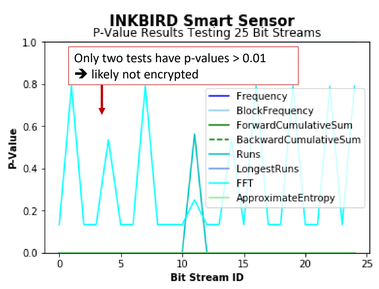
Encrypted file has ≈100% pass rate for all tests. Non-encrypted file has 0% pass rate for all tests

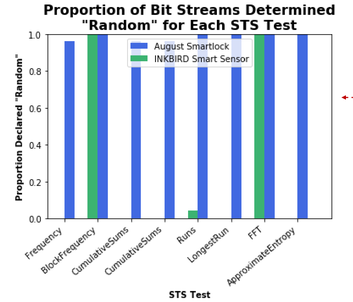


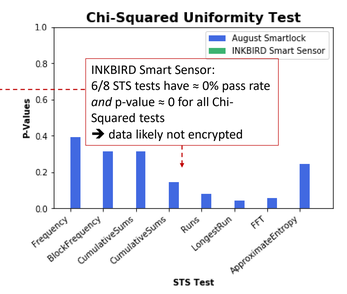


Testing Bluetooth ATT Data









Summary and Future Work

- Our results demonstrate that our approach is able to detect app-layer encryption.
- One limitation of our approach is that the STS is sensitive to the presence of structure inherent in packet headers and protocols. In future work, we will improve how ATT payload data is parsed to automatically detect and remove such structure.
- Our current results are limited to the ATT protocol, which is only used on BLE devices. Future work will examine other protocols (e.g., L2CAP) where data is transmitted for non-BLE devices.
- In future work, we will also expand our security audit to test additional BLE devices for application-layer encryption.

References

[1] Rizvi, Syed, et al. "Identifying the attack surface for IoT network." *Internet of Things* 9 (2020): 100162.
[2] Xu, Fenghao, et al. "BadBluetooth: Breaking Android Security Mechanisms via Malicious Bluetooth Peripherals." *NDSS*. 2019.
[3] National Institute of Standards and Technology Special Publication 800-22 revision 1a Natl. Inst. Stand. Technol. Spec. Publ. 800-22rev1a, 131 pages (April 2010).
[4] Marton, Kinga, and Alin Suciu. "On the interpretation of results from the NIST statistical test suite." *Science and Technology* 18.1 (2015): 18-32.
[5] OpenSSL. Online at https://www.openssl.org/
[6] August Smartlock Device. Online at https://august.com/
[7] INKBIRD Smart Sensor Device. Online at https://www.ink-bird.com/

This work was supported by the National Science Foundation Award CCF-1947750

[1] *Our Bluetooth dataset is in the process of being prepared for public release.*
[2] *Baseline encrypted file is a binary representation of '0123456789' repeated until 50,000 bits long, encrypted using OpenSSL's implementation of AES-128-CBC [5].*

MONTANA STATE UNIVERSITY