UNIVERSITY of WASHINGTON

EE 418 Project 2 – Clock-Based Intrusion Detection in Controller Area Network

Autumn 2017

Instructor: Xuhang Ying
Network Security Lab (NSL)
Department of Electrical Engineering

Project Guidelines

- Due **11:59pm, Dec 8 (Fri), 2017**
- Max. group size is **3**. Email instructor and TA if you want to change your group.
- Submit both **project report** and **source code** via Dropbox
  - Either MATLAB or Python.
  - Provide in-line comments to help understand your code, and make sure your code is ready to run.
- On the front page of your project report, provide
  - Names and student IDs of group members.
  - Clear description of each member's contribution.

Controller Area Network (CAN)

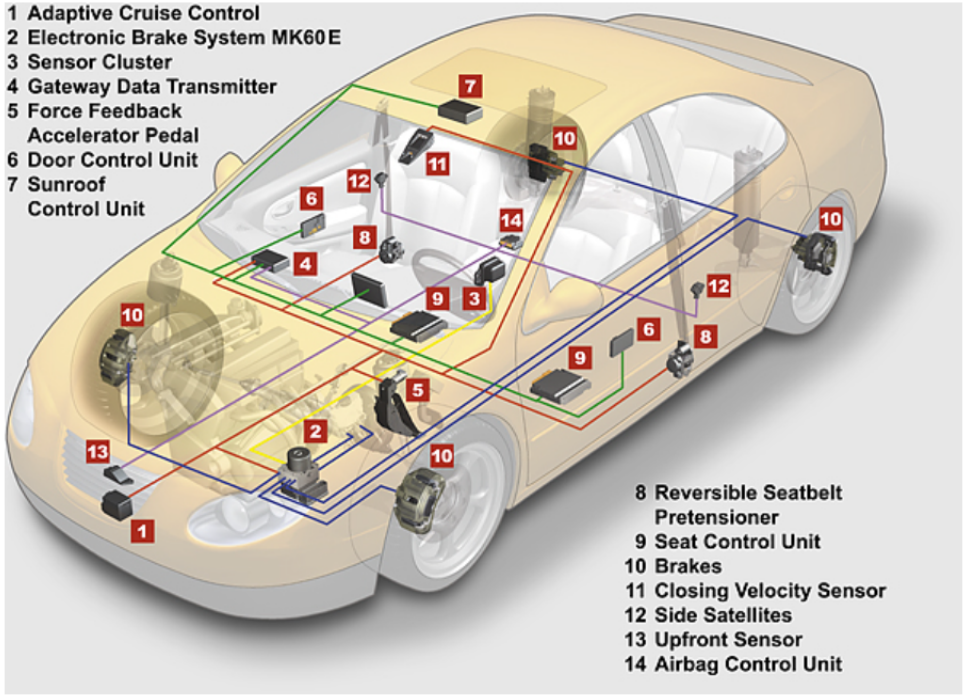


Figure – In-vehicle CAN bus

- Electronic Control Units (ECUs)
  - As many as 70, or even more
  - Independent computers
  - Engine, brakes, body control, radio, heating etc.
- CAN bus:
  - Allow ECUs to communicate
  - Bits are transmitted via voltage change on the bus
- In-vehicle CAN is critical for many functionalities

Vulnerabilities of CAN

- CAN is a broadcast bus: anyone can Tx and Rx packets to and from the bus.
- No authentication fields in the packet: cannot verify the origin of the message.
- Modern cars have many external interfaces (e.g., CD players or cellular radio), through which the adversary can compromise one or more ECUs.

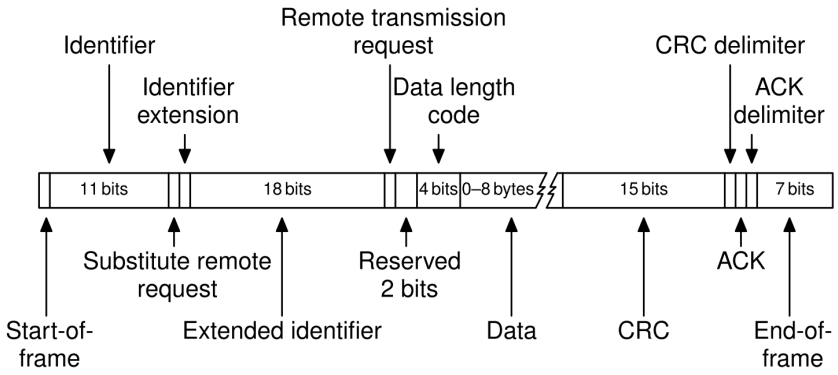


Figure – CAN packet structure

Attack Scenarios

- We consider an adversary who can
  - Physically/remotely compromise one or more in-vehicle ECUs;
  - Stop legitimate messages – **suspension** attack,
  - Inject spoofed messages – **fabrication** attack, or
  - Both at the same time – **masquerade** attack.

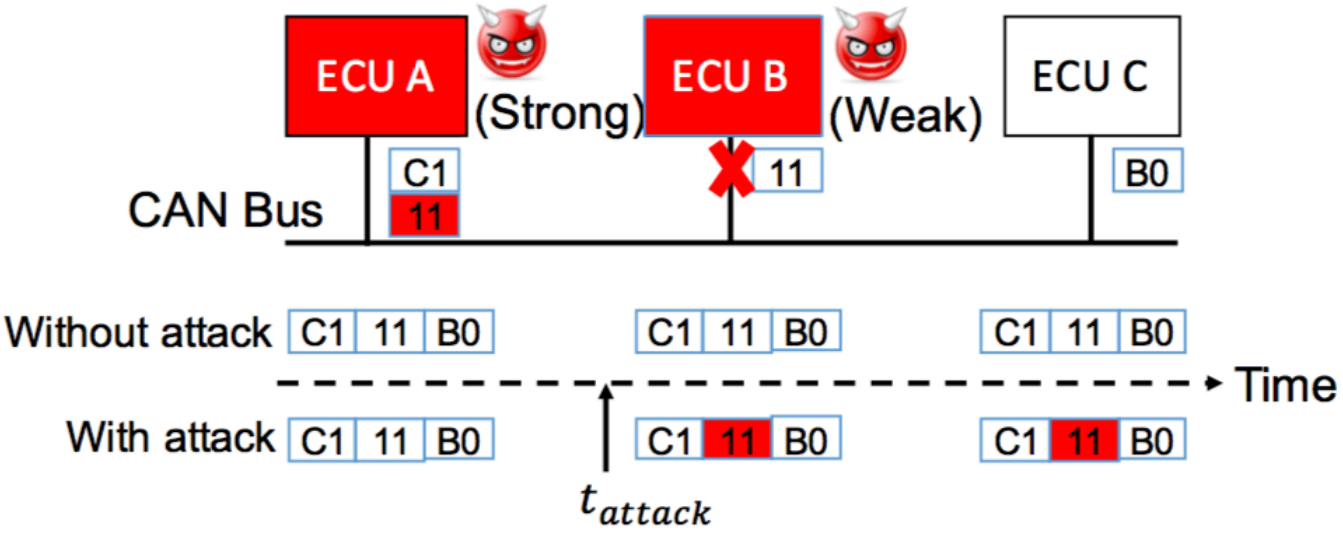
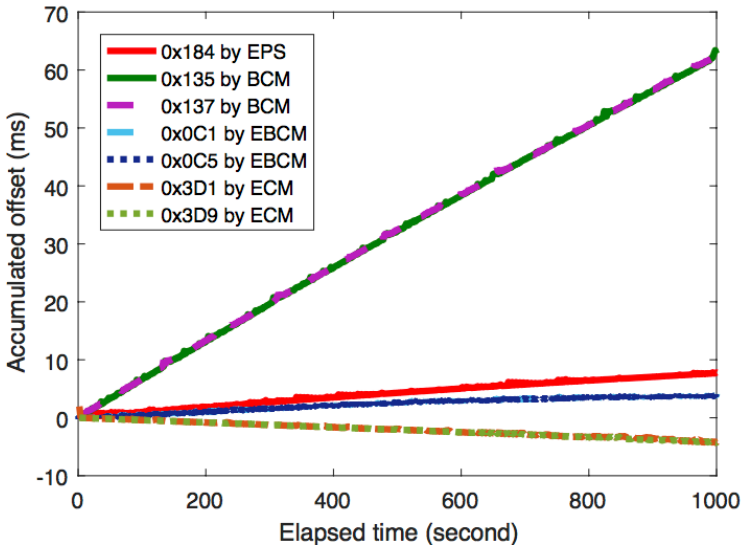
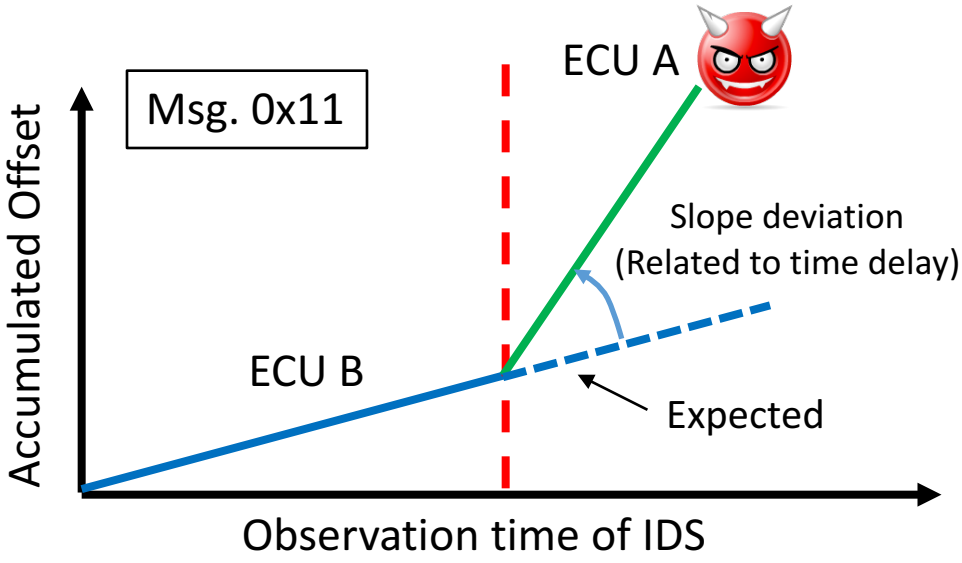


Figure – Illustration of masquerade attack.

Clock-Based Intrusion Detection System (IDS)

- State of the art: clock skew as unique identifier for each ECU.[1]



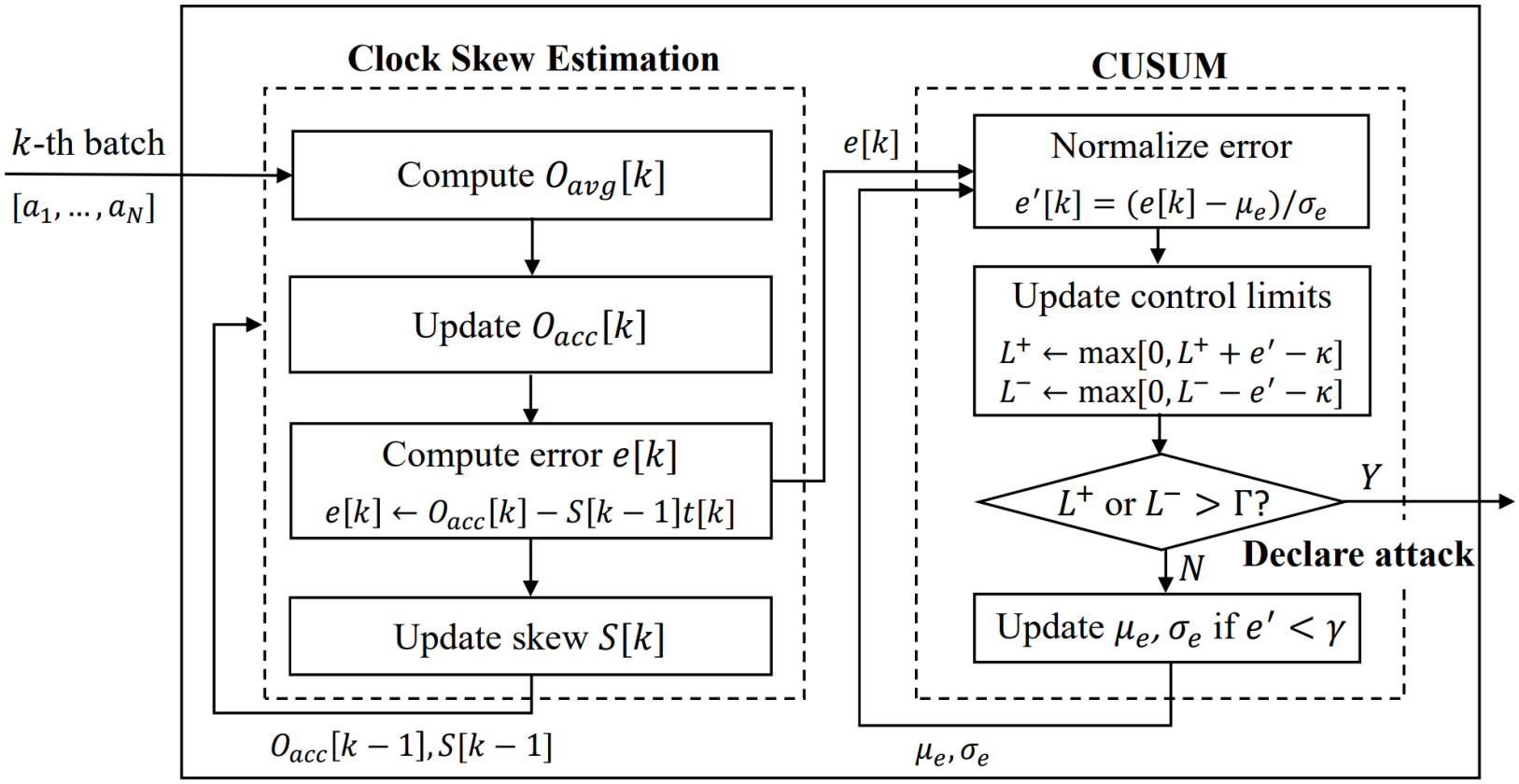


ECUs rely on local clocks that are not synchronized and run at different speeds (i.e., clock skew).

IDS learns and tracks the clock skew of the target message, and detects the masquerade attack.

[1] Cho, Kyong-Tak, and Kang G. Shin. "Fingerprinting Electronic Control Units for Vehicle Intrusion Detection." *USENIX Security Symposium*. 2016.

How IDS works



More Details

- Offset estimation:
  - Heuristic-based as in the state-of-the-art IDS [1] :

$$O_{avg}[k] = \frac{1}{N-1}\sum_{i=2}^{N}[a_i - (a_1 + (i-1)\mu_T[k-1])]$$

$$O_{acc}[k] = O_{acc}[k-1] + |O_{avg}[k]|$$

  - NTP-based as in the NTP-based IDS [2]:

$$O_{avg}[k] = T - \frac{a_N - a_0}{N}$$

$$O_{acc}[k] = O_{acc}[k-1] + N \cdot O_{avg}[k].$$

NTP: Network Time Protocol

[2] . S. Sagong, X. Ying, A. Clark, L. Bushnell, and R. Poovendran. “Cloaking the Clock: Emulating Clock Skew in Controller Area Networks”, Submitted to ACM/IEEE ICCPS 2018, Porto, Portugal, April 2018.

Intelligent Masquerade Attack – Cloaking Attack

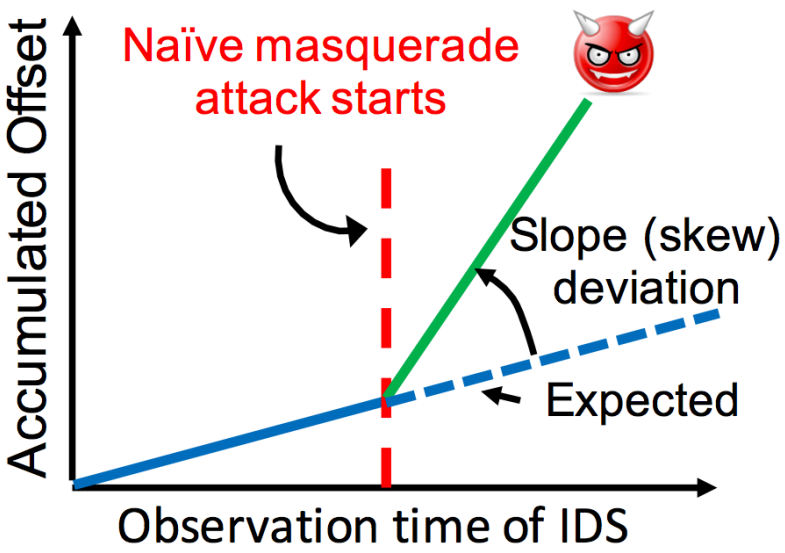


Figure – Naïve masquerade attack.

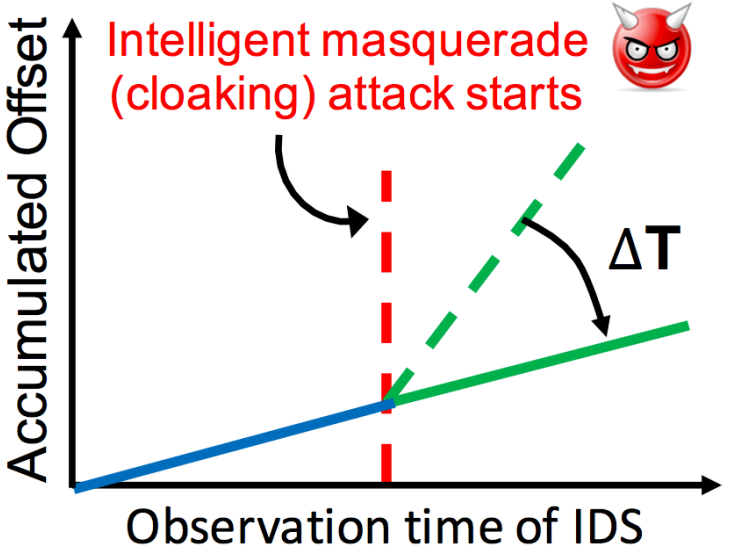


Figure – Intelligent masquerade attack.

Basic idea: An intelligent attacker can control the inter-departure times (by adding ΔT) to manipulate the clock skew estimated by IDS.

Your Assignment

- Dataset:
  - 184.txt, 3d1.txt, 180.txt
  - Three 10-Hz messages transmitted by different ECUs.
  - Each file contains arrival timestamps (for 50 mins).

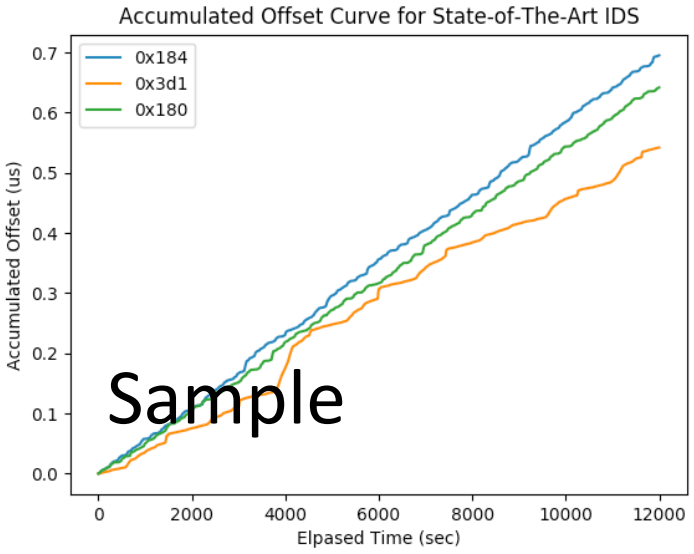
- Scenarios:
  - Scenario 1 – Masquerade attack: the adversary stops A's transmission of 0x184, and uses B to transmit spoofed message 0x184 every 0.1 sec (we treat 0x3d1 as 0x184).
  - Scenario 2 – Cloaking attack: the adversary stops A's transmission of 0x184, and uses B to transmit spoofed message 0x184 every (0.1 sec – 29us) (we treat 0x180 as 0x184).

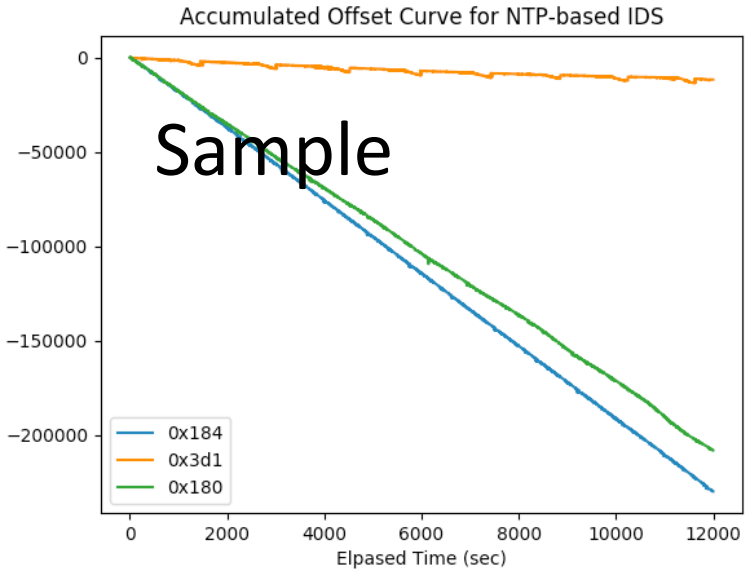
Source Code

- Python:
  - ids.py – implements class IDS (state-of-the-art and NTP-based).
  - simulation.py – main file that contains simulation code.
    - import_data(file=None)
    - plot_acc_offsets(ids, mode)
    - simulation_masquerade_attack(mode)
    - simulation_cloaking_attack(mode)
- MATLAB
  - IDS.m – implements class IDS
  - simulation.m – main file
  - import_data.m – import data from the txt file
  - plot_acc_offsets.m
  - simulation_masquerade_attack.m
  - simulation_cloaking_attack.m

Tasks 1 through 3

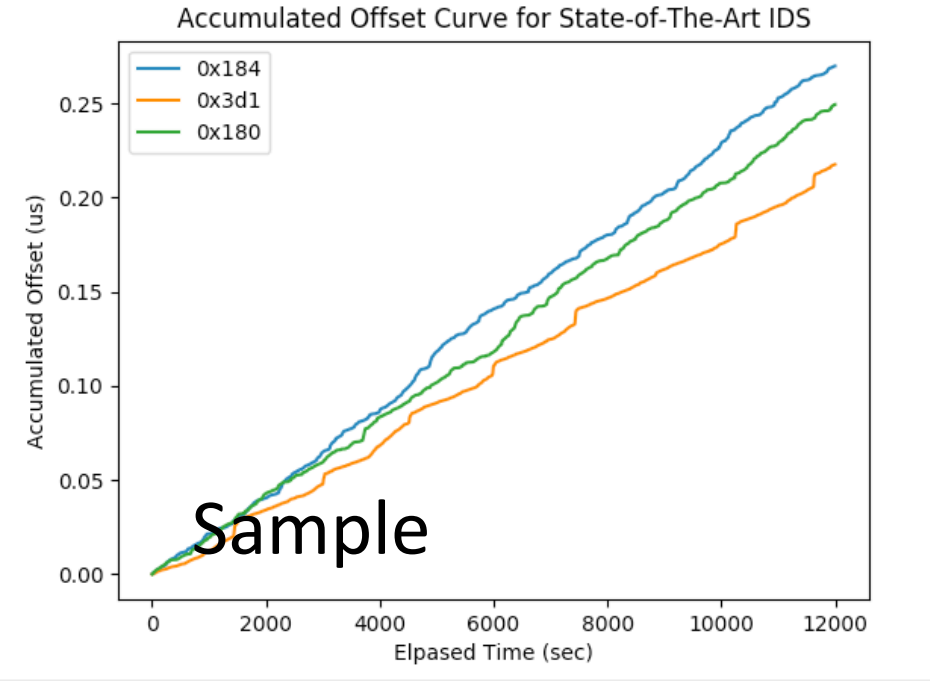
- Task 1: Implement class IDS
- Task 2: Plot accumulated offset curves as function of the elapsed time for the three messages (i.e., 0x184, 0x3d1 and 0x180) using the state-of-the-art IDS with batch size N = 20.
- Task 3: Repeat Task 2 for the NTP-based IDS with N = 20.

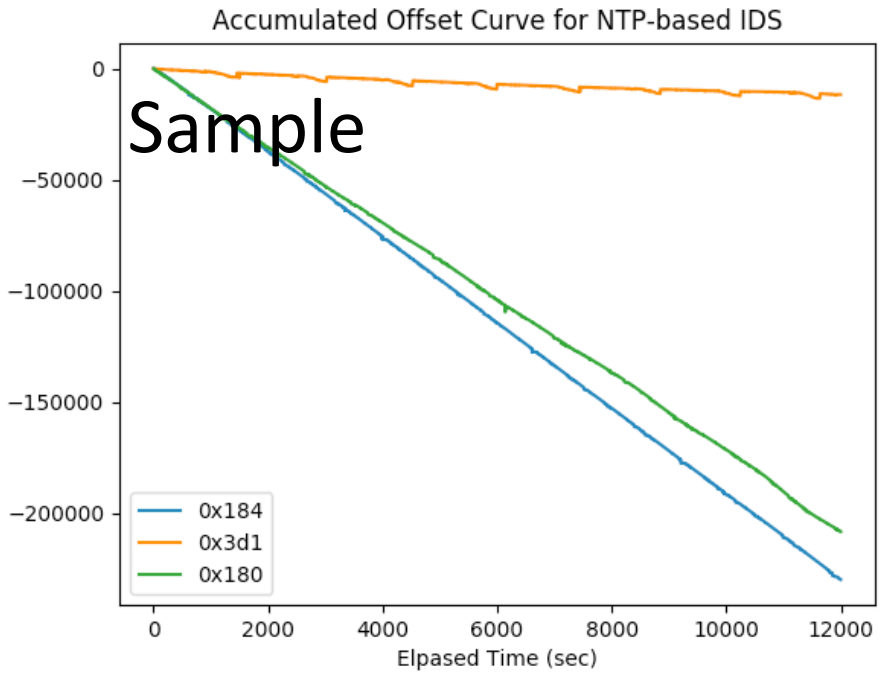




Task 4 – Impact of N

- Task 4: Repeat Tasks 2 and 3 with N = 30.

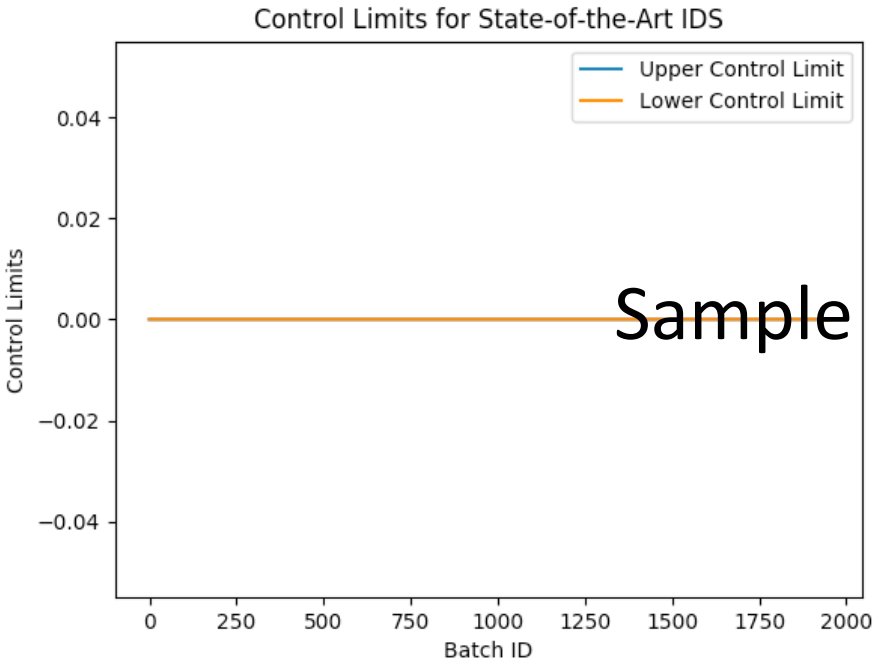


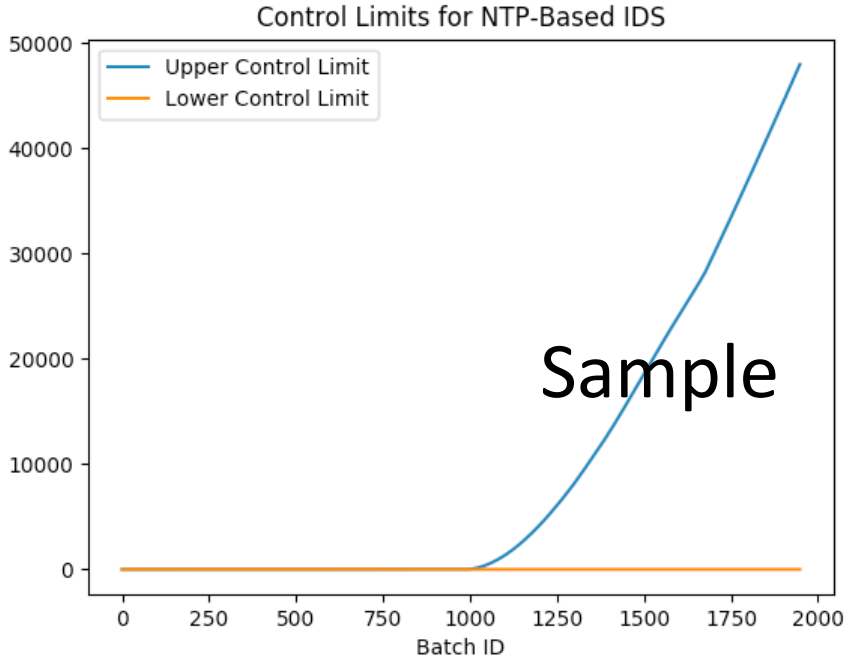


Comment on the consistency of clock skew estimation.

Task 5 – Masquerade Attack

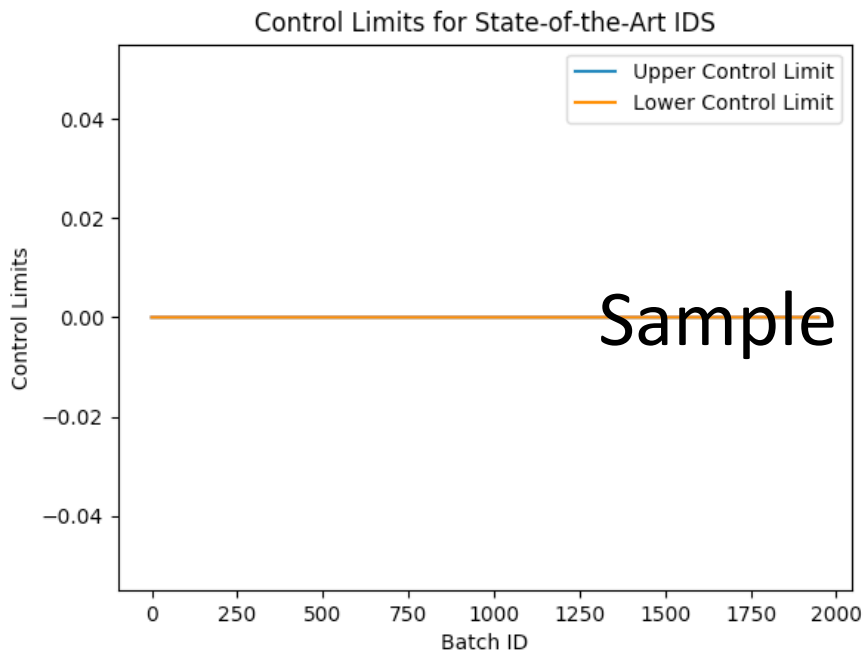
- Simulate the masquerade attack in Scenario 1, and plot control limits.

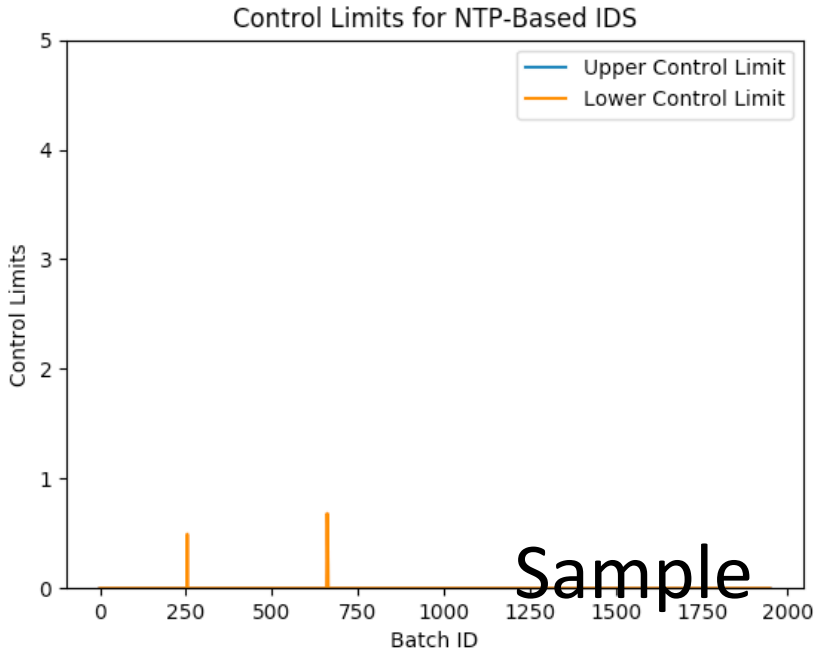




Task 6 – Cloaking Attack

- Simulate the cloaking attack in Scenario 2, and plot control limits.





Additional Questions

- Read [1] and [2], and answer 4 additional questions.

- Please include all necessary figures/plots, observations, and answers in your report.