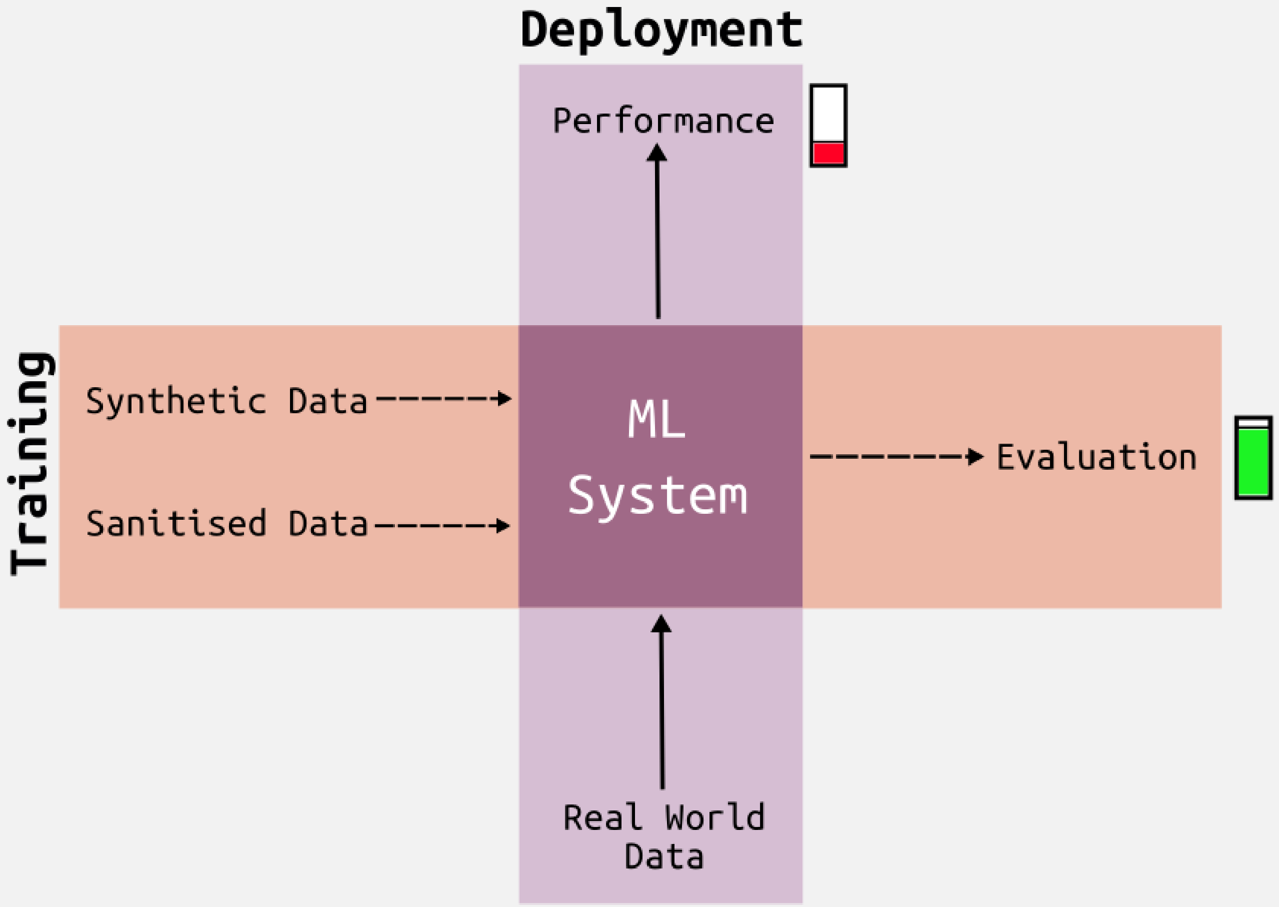
Data Generation and Sanitisation in Security-Sensitive Systems

Rob Flood

Background

- In certain domains, it is extremely difficult to train machine learning systems using datasets drawn directly from real-world distributions
- Particularly true for security applications of ML — privacy concerns for individuals and organisations
- Public benchmark datasets consist of artificially generated or (heavily) redacted data
- Difficulty in obtaining data limits classifier robustness due to need for constant updates
- Challenges:
    1. Evaluating the quality of synthetic `data
    2. Generating synthetic data
    3. Sanitising data whilst maintaining utility



Deployment

Performance

Training

Synthetic Data

Sanitised Data

ML System

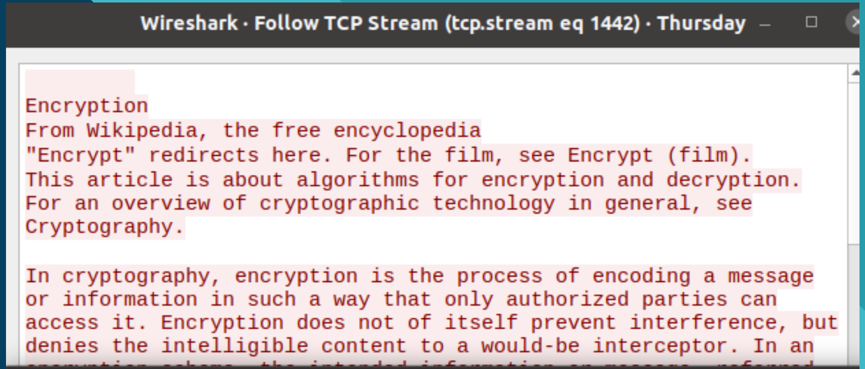
Evaluation

Real World Data

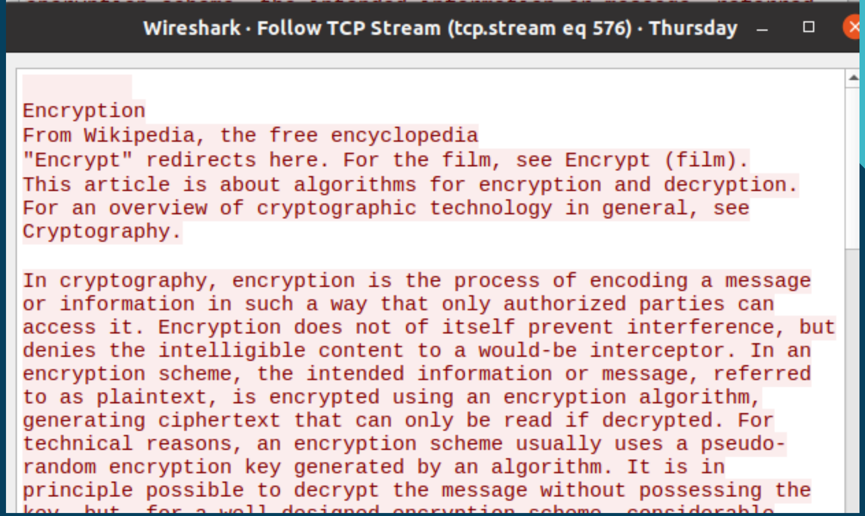
Δ(Performance, Evaluation) ~ Δ(Synthetic/Sanitised, Real World)

Evaluating Synthetic Network Traffic Datasets

- DARPA ‘98, KDD Cup ’99 spurred research into intrusion detection
- Superseded by NSL-KDD and then by CIC-IDS ‘17, UNSW NB15
- These datasets still have obvious flaws:
    - Lack of traffic variety
    - Poor attack realism
    - Simulation artifacts
    - Shoddy construction
- Currently, attempting to systematise a methodology for evaluating the quality of network traffic datasets

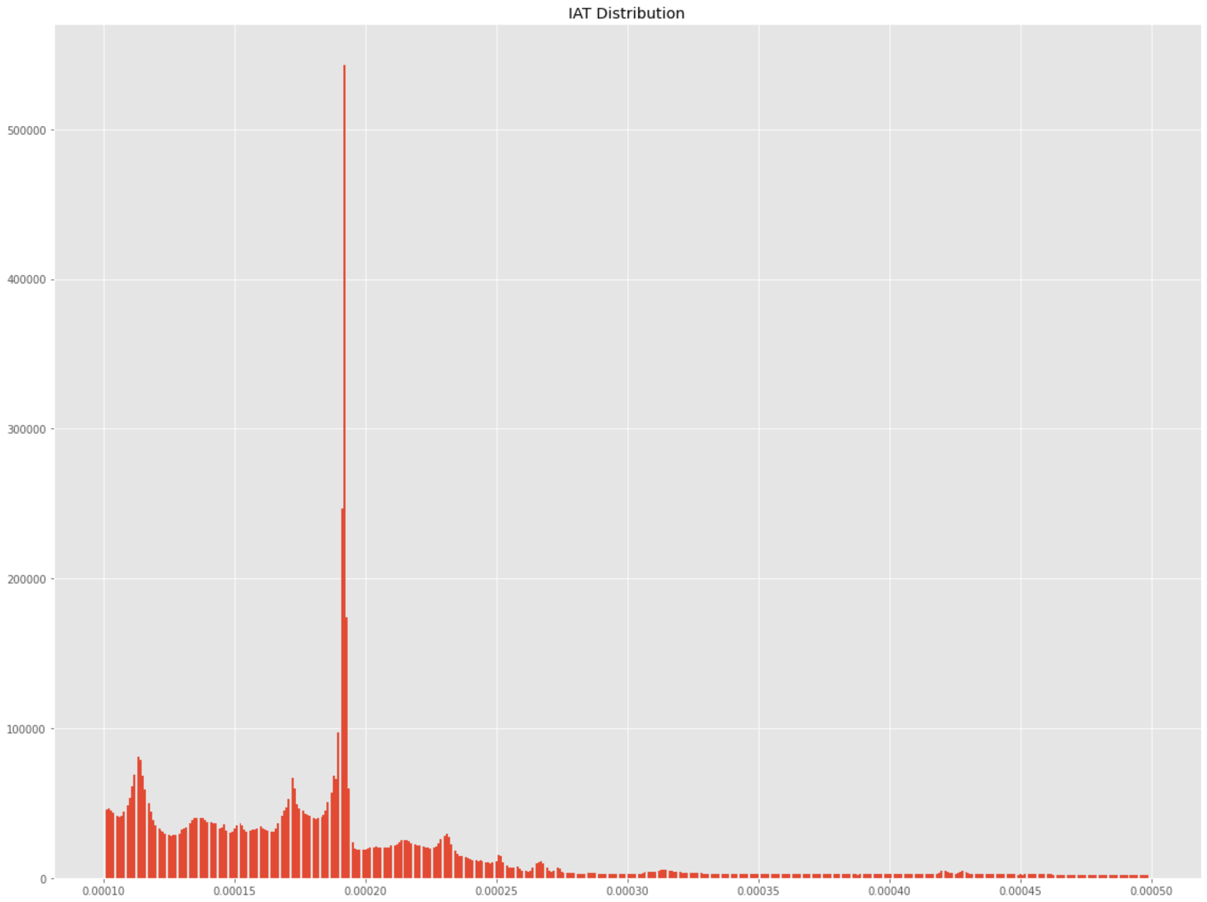

Wireshark · Follow TCP Stream (tcp.stream eq 1442) · Thursday
–
□
×

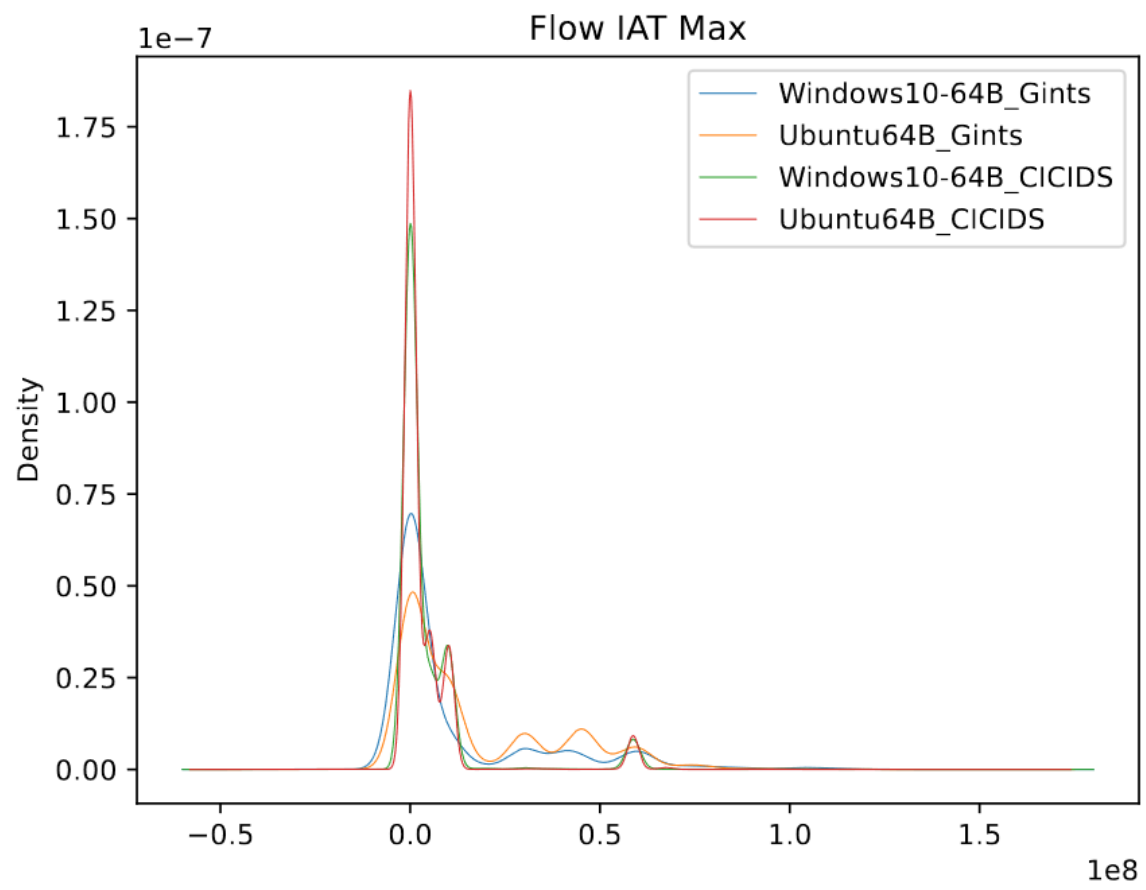
```
Encryption
From Wikipedia, the free encyclopedia
"Encrypt" redirects here. For the film, see Encrypt (film).
This article is about algorithms for encryption and decryption.
For an overview of cryptographic technology in general, see
Cryptography.

In cryptography, encryption is the process of encoding a message
or information in such a way that only authorized parties can
access it. Encryption does not of itself prevent interference, but
denies the intelligible content to a would-be interceptor. In an
```


Wireshark · Follow TCP Stream (tcp.stream eq 576) · Thursday
–
□
×

```
Encryption
From Wikipedia, the free encyclopedia
"Encrypt" redirects here. For the film, see Encrypt (film).
This article is about algorithms for encryption and decryption.
For an overview of cryptographic technology in general, see
Cryptography.

In cryptography, encryption is the process of encoding a message
or information in such a way that only authorized parties can
access it. Encryption does not of itself prevent interference, but
denies the intelligible content to a would-be interceptor. In an
encryption scheme, the intended information or message, referred
to as plaintext, is encrypted using an encryption algorithm,
generating ciphertext that can only be read if decrypted. For
technical reasons, an encryption scheme usually uses a pseudo-
random encryption key generated by an algorithm. It is in
principle possible to decrypt the message without possessing the
key, but, for a well-designed encryption scheme, considerable
```

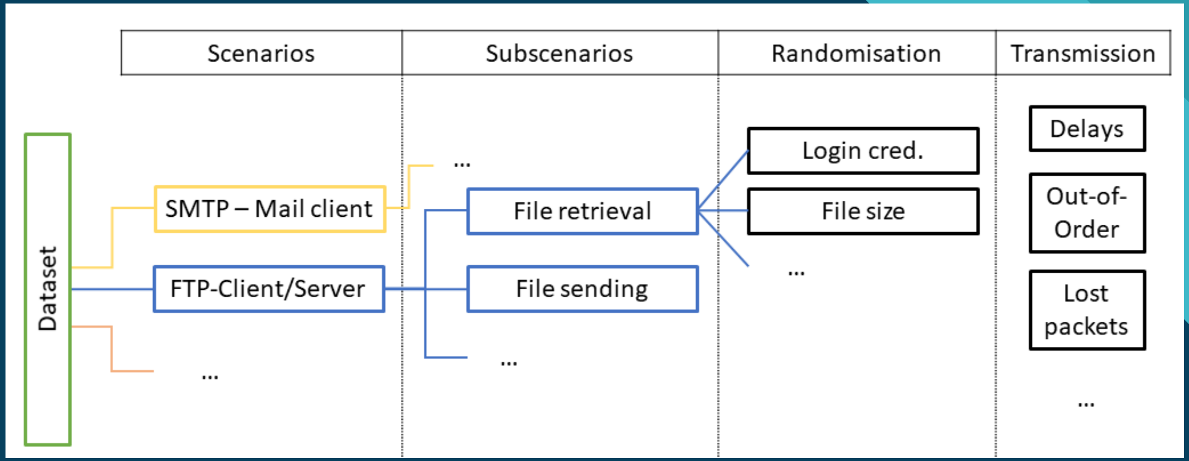
1


IAT Distribution
500000
400000
300000
200000
100000
0
0.00010
0.00015
0.00020
0.00025
0.00030
0.00035
0.00040
0.00045
0.00050


Flow IAT Max
1e−7
Density
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.5
0.0
0.5
1.0
1.5
1e8
Windows10-64B_Gints
Ubuntu64B_Gints
Windows10-64B_CICIDS
Ubuntu64B_CICIDS

Network Data Generation - DetGen

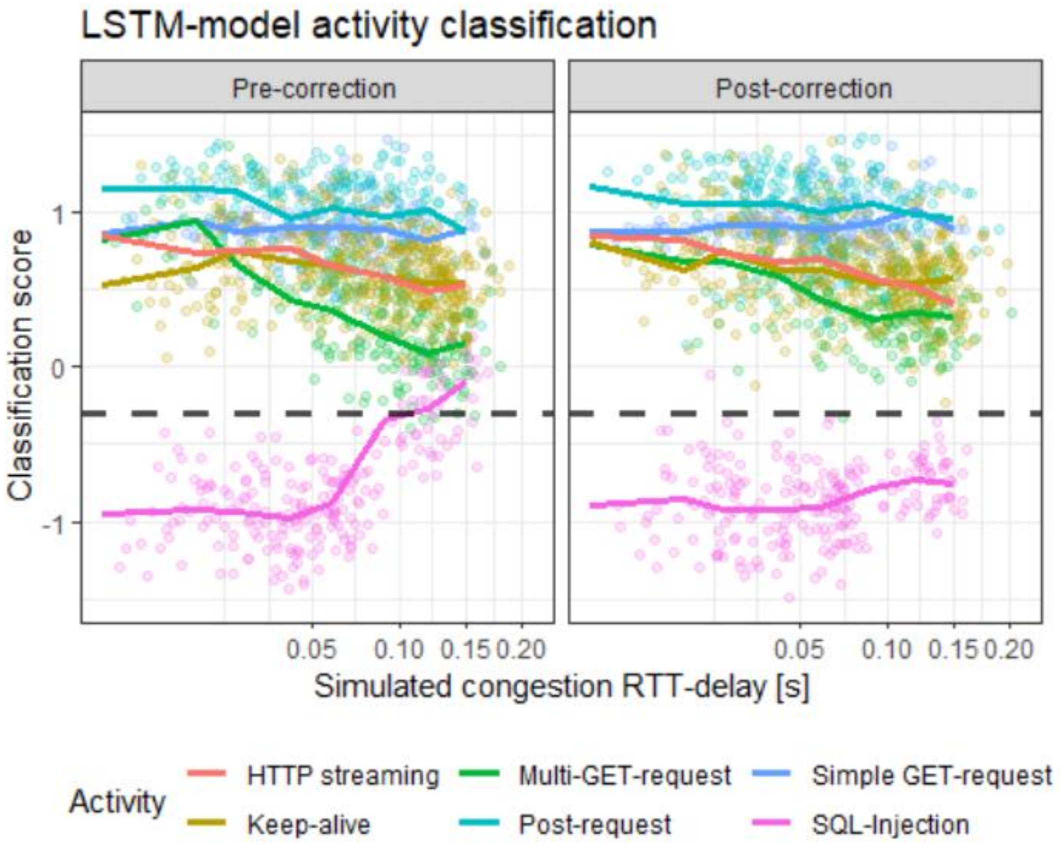
- DetGen – ‘Deterministic’ network traffic generation using containers
- Can generate traffic with accurate ground truth with control over many traffic features
  - Protocol
  - Congestion
  - Packet loss
  - Corruption
  - Duplication
- Have seen success in using DetGen to produce realistic network traffic[1]
- Currently, porting to Mininet for realistic topology emulation; chaining together scenarios

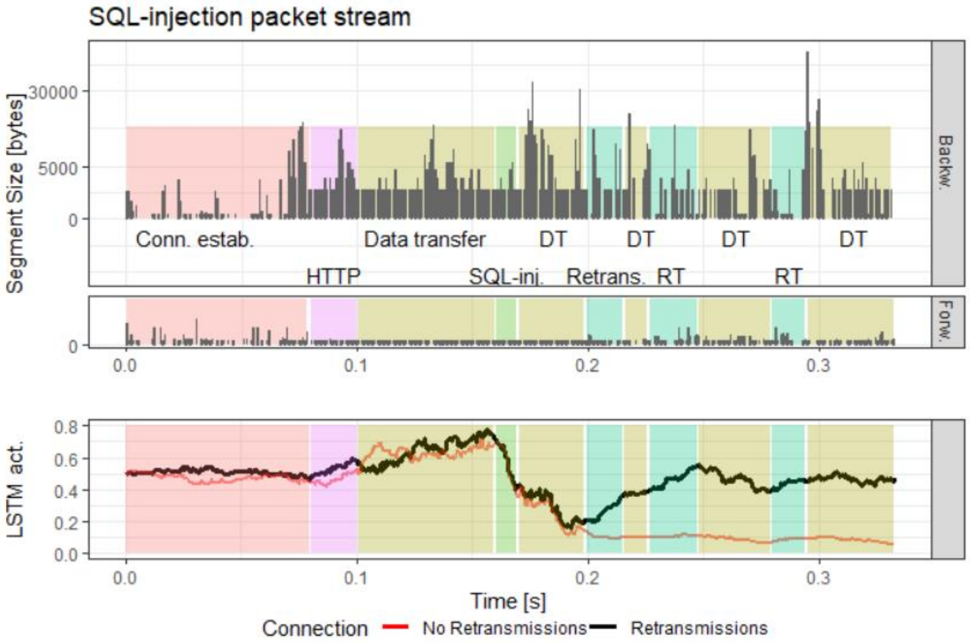


[1] Traffic generation using containerization for machine learning. Dynamics 2019, Henry Clausen, Robert Flood & David Aspinall

2

2

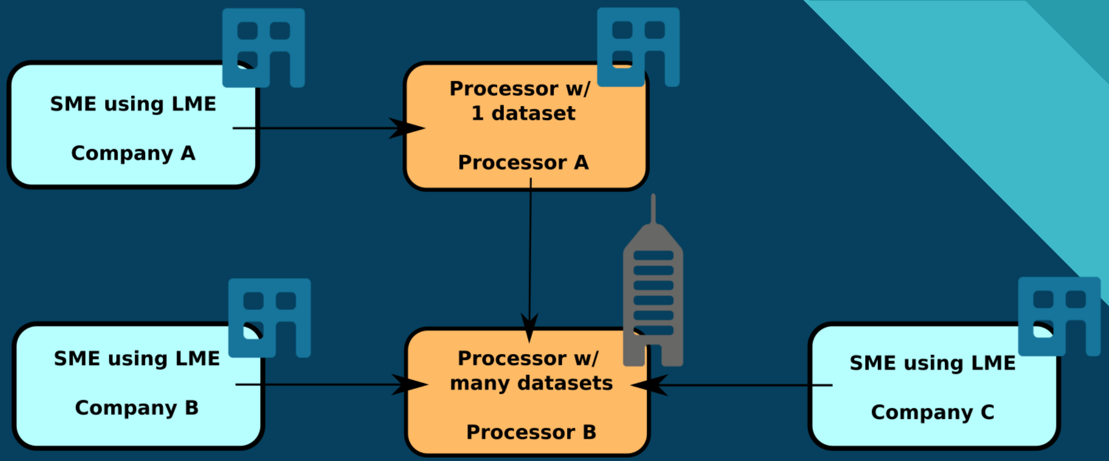

LSTM-model activity classification


SQL-injection packet stream

2: Controlling network traffic microstructures for machine-learning model probing, SecureComm 2021, Henry Clausen, David Aspinall & Robert Flood

Data Sanitisation

- GDPR/DPA recommend certain pseudonymisation/anonymisation methods such as k-anonymity/differential privacy
- Domain experts often choose what data to obfuscate on an ad hoc basis
- Questions emerge when multiple parties with differing privacy policies interact with one another
- Want to encapsulating this process as an ‘Anonymisation Policy’ – collation of data, data shared amongst multiple parties …
- Want these policies to have certain properties: composition, hierarchy …


SME using LME

Company A

Processor w/ 1 dataset

Processor A

SME using LME

Company B

Processor w/ many datasets

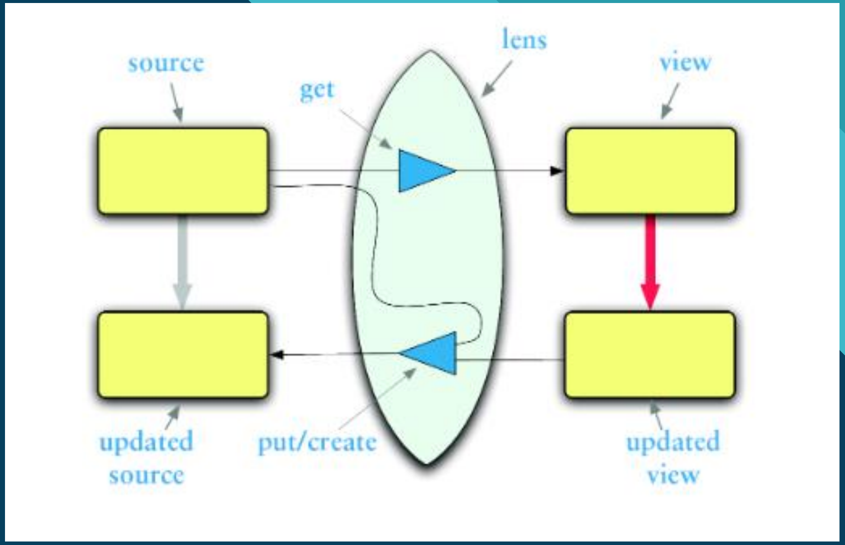
Processor B

SME using LME

Company C

Data Sanitisation – AnonLens (WIP)

- Idea: Given an operation from a Database to a ‘View’ (*get*), automatically derive a reverse operation mapping a View to a Database (*put*) – a lens
- At a high-level, similar to the problem of producing many anonymised versions of some source data – treat anonymisation functions as *gets*
- Maintain consistency across a variety of views thanks to lens laws
- Can be easily expressed, composed in manner that maintains lens laws
- Modification of data explicitly defined in a functional manner
    - Reverse operation (deanonymisation) easy to derive in a fully auditable manner
    - Many of the measurements we need to derive policy properties gotten for ‘free’



Thank You