Privacy-Preserving Tampering Detection in Automotive Systems

Adrian-Silviu Roman, Béla Genge , Adrian-Vasile Duka and Piroska Haller

MDPI Electronics, 2021

Presented by: Kyle Trevis

Purpose

- Modern Automobiles record and process large amounts of sensitive data
- Tampering involves targeted manipulation of data
  - Attacks on drivers
  - Financial gain (Odometer, Emission Control)
- Tampering Detection must be done outside the vehicle while preserving data privacy

Existing Techniques

- Encryption, Anonymization, and **Perturbation**
  - Randomization and Transformation Based
- Data Transformation can allow for low complexity, high privacy, and preservation of Euclidean data
- Several forms of Data Transformation exist, this paper focuses on Fast Fourier Transform (FFT)
  - O(n log n)

Privacy Preservation Technique	Computation Operations	Privacy Preservation Location	Applicable on Multiple Sensors Simultaneously	Adjustable Level of Privacy	Computation Complexity
Lightweight homomorphic encryption [13]	additive and multiplicative homomorphic encryption	on an external trusted server	no	no	high
PPMDS [50]	additive homomorphic encryption and signing	locally	no	no	medium
FFT-based data perturbation	data transformation, frequency filtering, noise addition	locally	yes	yes	low

Tampering Detection Pipeline

Performing FFT and Filtering

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

- N: Number of discrete values per sensor
- M: Number of sensors
- *u*,*v*: Frequency Values
- *x*,*y*: Time domain values

Performing FFT and Filtering

- *A* matrix contains original sensor data
  - Transformed into ***A*** = ***M*** x ***N***, where ***M***,***N*** = multiple of 2
  - Dominant Component F(0,0) is centered in the matrix
  - Rest of matrix padded with zeros

$$\bar{F}(u,v) = H(u,v) \cdot F(u,v)$$

$$H(u,v) = \begin{cases} 1, & \text{if } \sqrt{u^2+v^2} \leq f_c \\ 0, & \text{otherwise,} \end{cases}$$

- Matrix *F* represents all transformations applied in frequency domain
  - Gaussian noise, preliminary filtering, etc.
  - ***F*** is computed through application of final Ideal 2D Low-Pass Filter

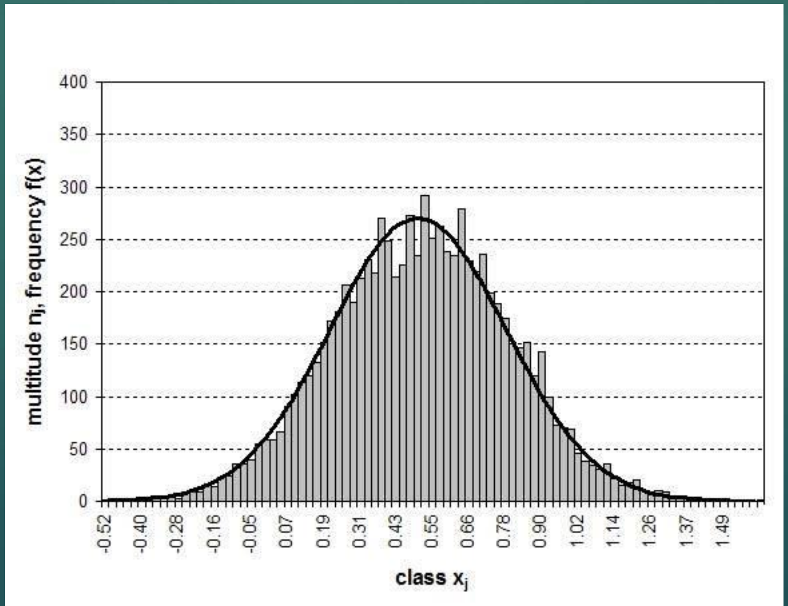
Algorithm 1: FFT-based data distortion.

```
Input: A (Sensor data); f_c (Cut-off frequency); σ (Noise variance)
Output: D (The distorted data)
Function ComputeDistortedData(A, f_c, σ):
    [M, N] ⟵ size(A);
    Ā ⟵ zeropadding(A); // Zero pad to the next power of 2
    [M̄, N̄] ⟵ size(Ā);
    Â ⟵ Ā;
    for x ← 1 to M̄ do
        for y ← 1 to N̄ do
            Â(x, y) ← Ā(x, y) · e^{πi(x+y)};
        end
    end
    F ⟵ FastFourierTransform(Â);
    H ⟵ ComputeFilter(f_c, M̄, N̄); // Get the filter matrix H
    F̄ ⟵ H · F; // Apply the filter
    F̄_+ ⟵ AddGaussianNoise(F̄, σ); // Add Gaussian white noise
    D̄ ⟵ InverseFastFourierTransform(F̄_+); // Get the distorted data
    D ⟵ crop(D̄, M, N); // Get only data from the top-left corner
    return D
End Function
```

Adding Gaussian White Noise

- Discord value σ represents the magnitude of distortion the data set can handle before data is irretrievable
  - Allows for easily tunable levels of privacy

- Perturbation process with variance σ^2 preserves a signals properties



[2]

Algorithm 2: Add Gaussian white noise to the frequency matrix.

```
Input: F̄ (The filtered frequency matrix); σ (Noise variance)
Output: F̄+ (The distorted frequency matrix)
Function AddGaussianNoise(F̄,σ):
    [M̄, N̄] ⟵ size(F̄);
    N+ ⟵ sum(F̄ > 0); // Get the number of frequencies > 0,
    K ⟵ sum(abs(F̄) ≥ σ); // and the number with magnitude > σ
    for i ← 1 to M̄ do
        for j ← 1 to N̄ do
            if abs(F̄(i,j)) ≥ σ then
                F̄+(i,j) ⟵ F̄(i,j) + GaussRnd(0, σ/2 √(N+/K))(1 + i);
            else
                F̄+(i,j) ⟵ F̄(i,j);
            end
        end
    end
    return F̄+ // Return the distorted frequency matrix
End Function
```

Tampering Detection

- Combines use of Random Forest (RF) and Univariate Cumulative Sum (UCUSUM)
  - Regression and gradual change of monitored data

- Analyzing detects anomalies indicative of tampering

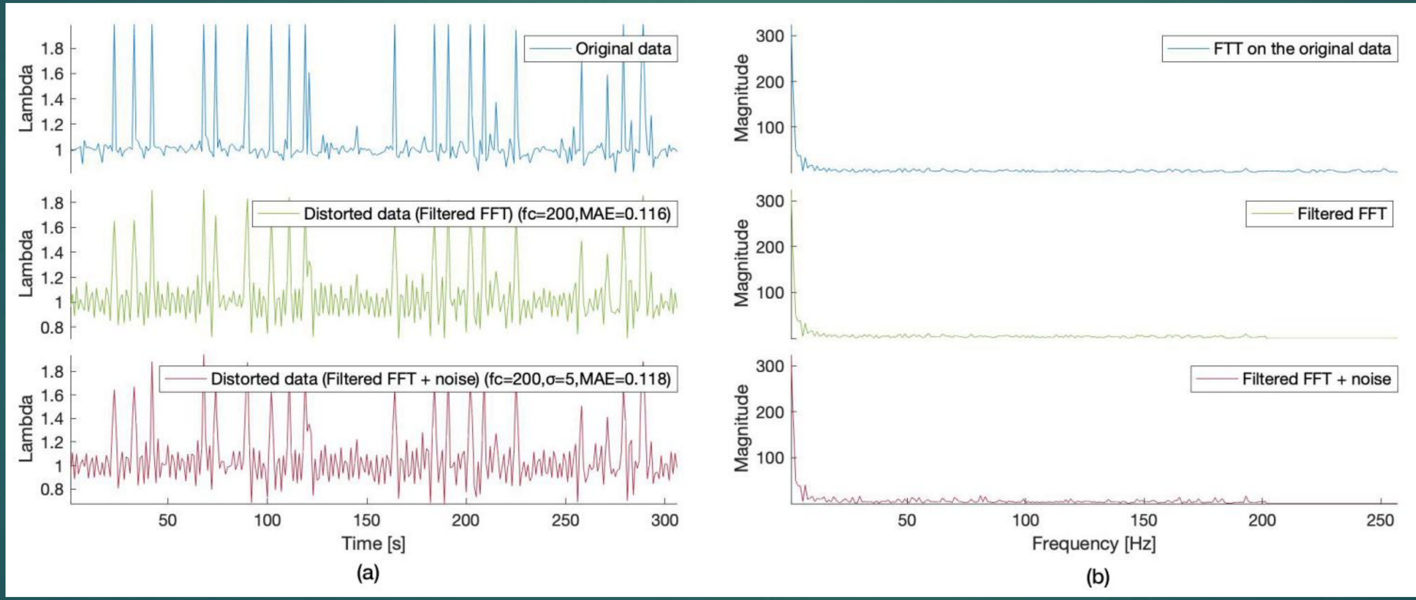
- For testing purposes, True and False Positive rates were computed
  - True Positive: Data properly detected as an anomaly
  - False Positive: Data that is not an anomaly but detected as such
  - False Negative: Data that is an anomaly, but not detected

$$TPR = \frac{TP}{TP + FN}.$$

$$FPR = \frac{FP}{FP + TN}.$$

Results

- Test One: 1D Sensor Data FFT-Based Distortion
  - Data collected from On Board Diagnostic (OBD) II oxygen sensor
  - 2015 EUR6 Skoda Rapid 1.2 L TSI passenger vehicle
  - Test used to prove validity of FFT data transform and added distortion



Results

- Test Two: 2D Sensor Data FFT-Based Distortion
  - Oxygen sensor, Oxygen jump sensor voltage, Engine torque, Throttle position, and Coolant temperature all recorded
  - Test used to prove computation complexity

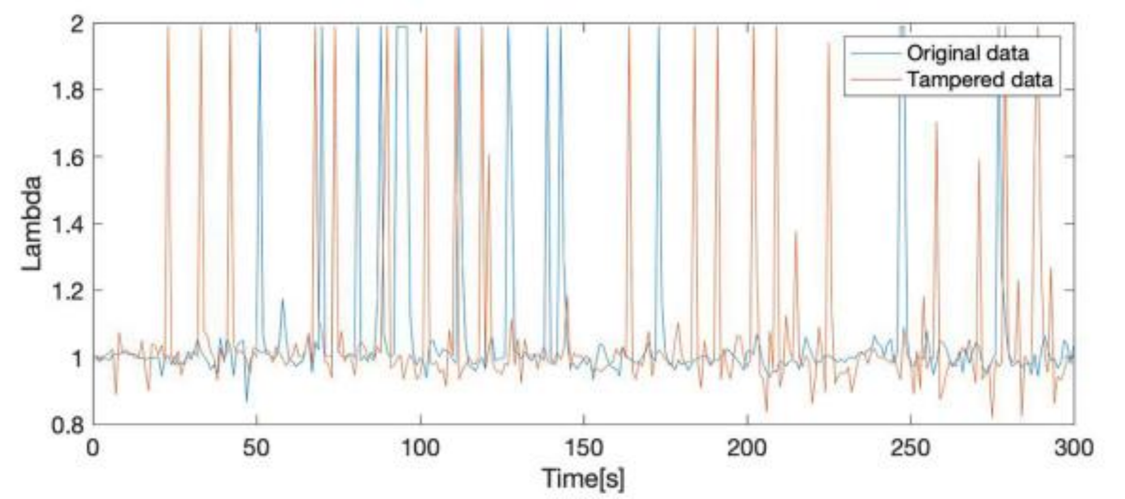
No. of Sensors	Overall Exec. Time (ms)	Exec. Time/Sensor (ms)	Data Reduction (%)
1	9.5	9.5	34.3
2	9.7	4.9	34.6
3	9.9	3.3	13.0
5	10.2	2.0	−4.18
10	11.1	1.1	−4.11
12	12.6	1.0	13.23

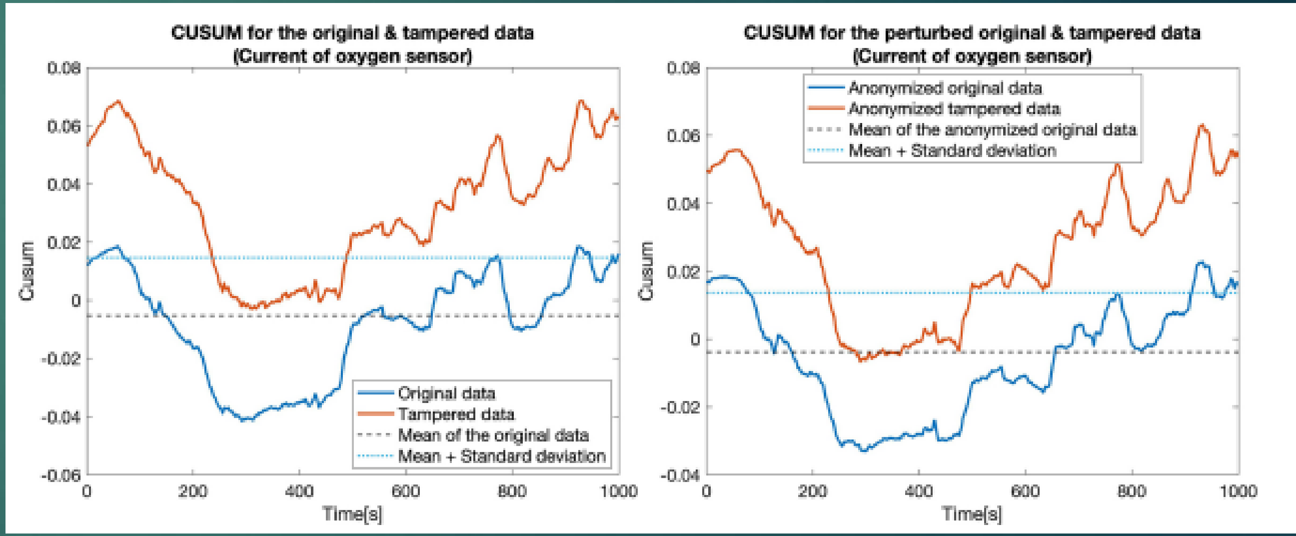
Results

- Test Three: Privacy-Preserving Tampering Detection
  - Test used to prove anomalies can still be detected after transformation
  - New data set recorded with similar conditions to previous tests
  - Tampered by substituting values from previously recorded data sets

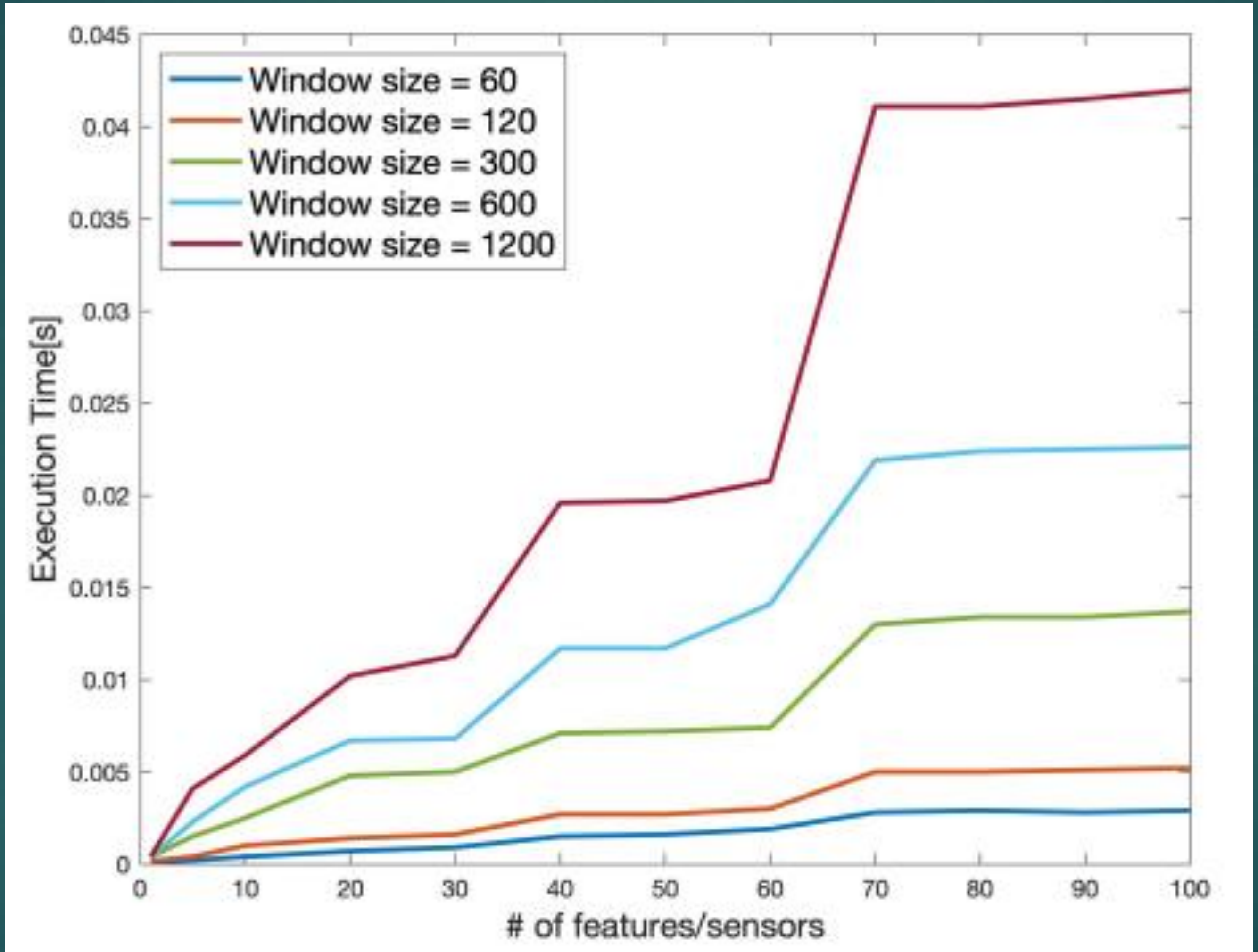
Results

Current of Oxygen Sensor Data





# of Tampered Sensors	Tampered Sensor(s)	Clear Data TPR	Clear Data FPR	Anonymized Data TPR	Anonymized Data FPR
1	Current of oxygen sensor	77.4%	18.5%	76%	21.5%
1	Oxygen jump sensor voltage	100%	18.5%	100%	21.5%
1	Coolant temperature	100%	18.5%	100%	21.5%
1	Throttle valve position	100%	18.5%	100%	21.5%
1	Engine torque	100%	18.5%	82.7%	21.5%
2	Current of oxygen sensor, Oxygen jump sensor voltage	100%	18.5%	100%	21.5%
2	Current of oxygen sensor, Engine torque	100%	18.5%	99.4%	21.5%
2	Engine torque, Coolant temperature	100%	18.5%	100%	21.5%
2	Engine torque, Throttle valve position	87.4%	18.5%	100%	21.5%
4	Current of oxygen sensor, Coolant temperature, Engine torque, Throttle valve position	100%	18.5%	100%	21.5%


Window size = 60
Window size = 120
Window size = 300
Window size = 600
Window size = 1200
Execution Time[s]
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
10
20
30
40
50
60
70
80
90
100
# of features/sensors

Contributions

- FFT is an effective technique for privacy preserving tampering detection
  - Retains data characteristics for anomaly detection
  - Scalable levels of privacy
  - Low complexity cost compared to existing methods

- Synthesis of RF and UCUSUM result in effective tampering detection
  - Exhibits up to 100% detection rate
  - False Positive rate of 21% suggests further improvement

Future Work

- Tampering Detection can be improved
  - Reduce False Positive Rate
- Further testing in embedded environment
- Real-time execution and pre-processing

My Thoughts

- The creative use of FFT for data transformation is both novel and effective
- 100% detection rate for tampering detection
- Paves the way for future work in this field
- Not much evidence that FFT properly obscures data from privacy attack

Discussion

- Do you think that FFT transforms, filtering, and Gaussian White Noise can be safely assumed to protect data?

- Is FFT or other data transformation used in other autonomous systems? What are some more applications in autonomous systems?

- Are there additional advantages/disadvantages of data transformation compared to Encryption/Anonymization not covered in this paper?

Cited Works

- Roman, A.-S.; Genge, B.; Duka, A.-V.; Haller, P. PrivacyPreserving Tampering Detection in Automotive Systems. Electronics 2021, 10, 3161

- Flegner, Patrik & Ján, Kačur & Durdan, Milan & Marek, Laciak. (2015). Application of adaptive filters in rock separation by rotary drilling process identification. Acta Montanistica Slovaca. 20. 38-48. 10.3390/ams20010038.