A Privacy Preserving Approach to Energy Theft Detection in Smart Grids

Christopher Richardson and Nicholas Race

School of Computing & Communications
Lancaster University, UK

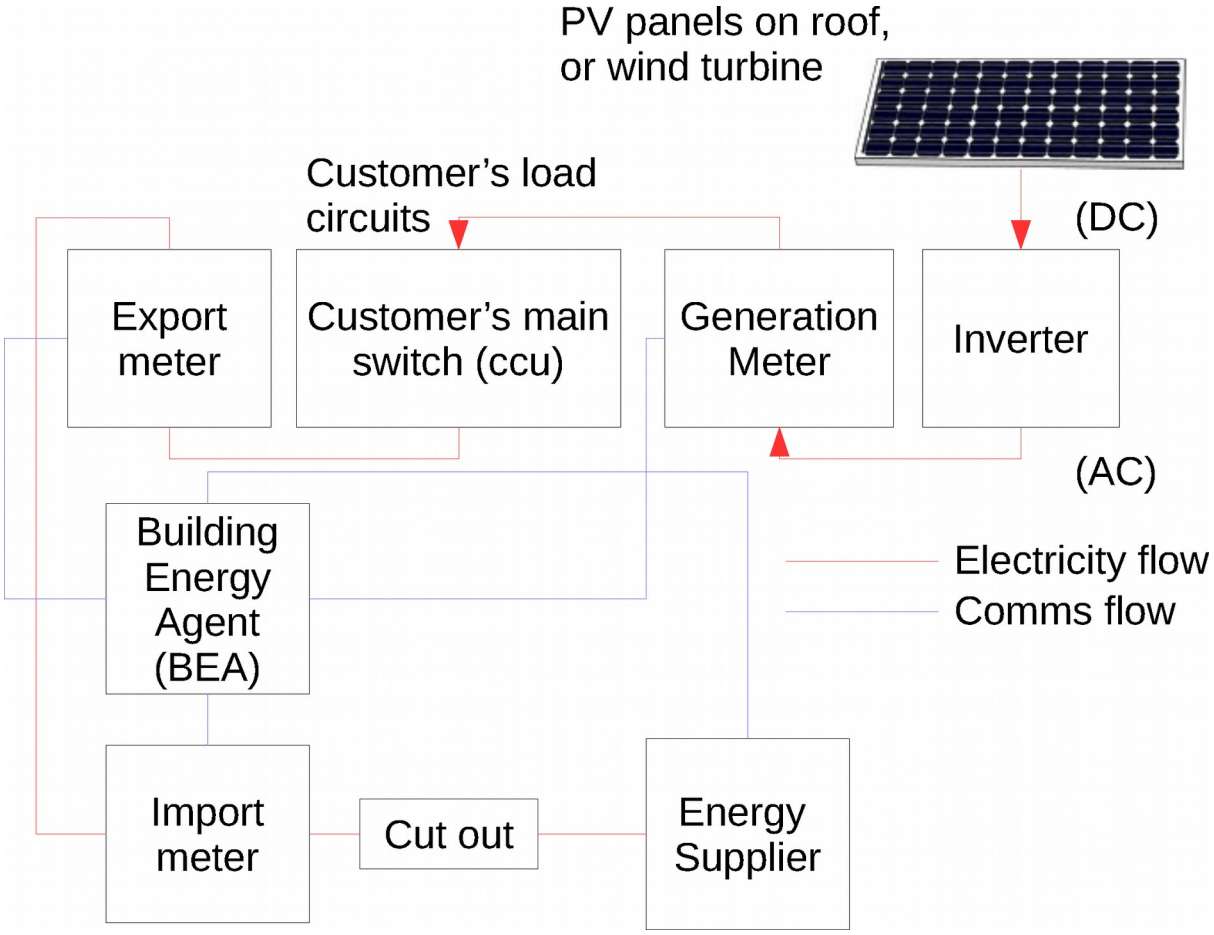
c.richardson@lancaster.ac.uk,
n.race@lancaster.ac.uk

Paul Smith

Digital Safety & Security Department
AIT Austrian Institute of Technology
Vienna, Austria

paul.smith@ait.ac.at

Aim to detect tampering of smart meters for financial gain



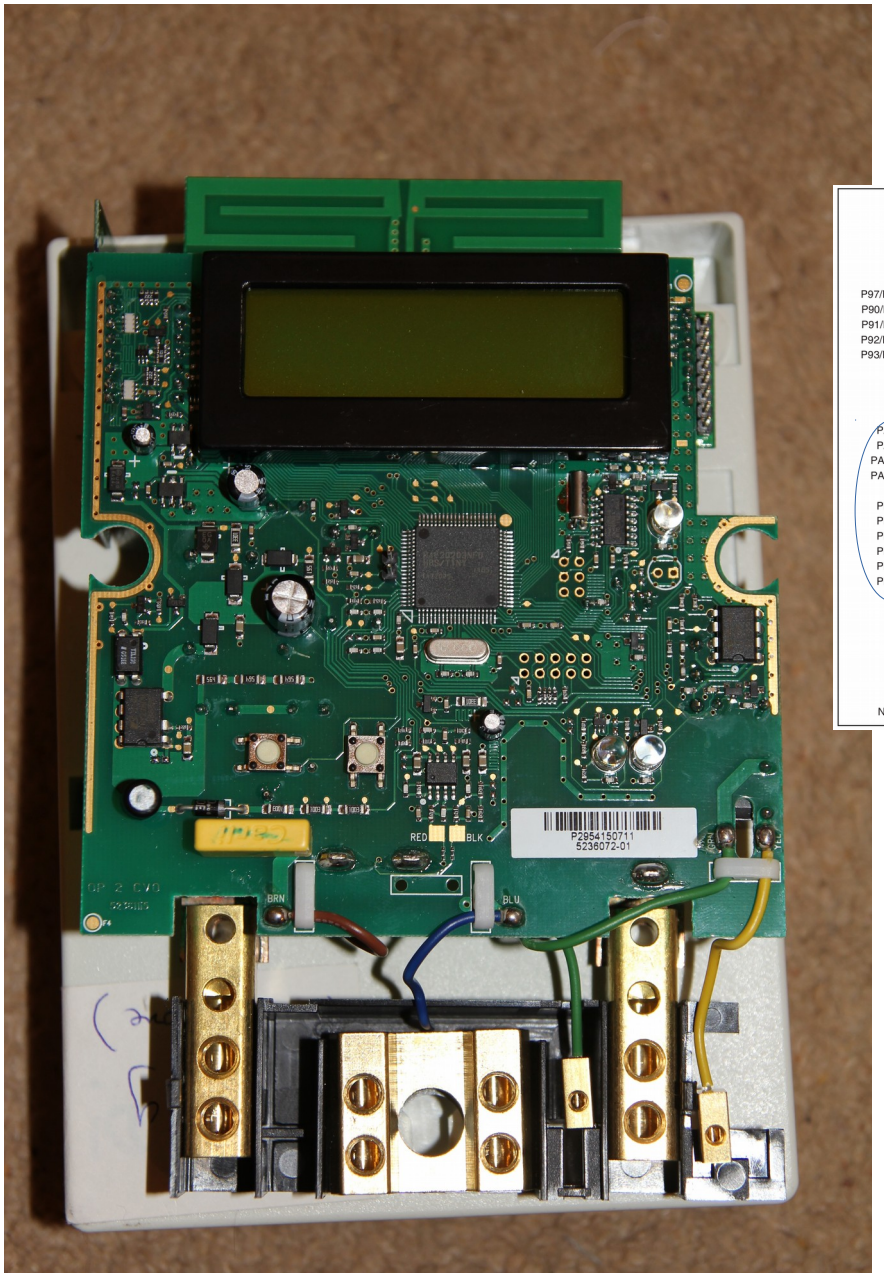
Based on a diagram by Tom Chevalier

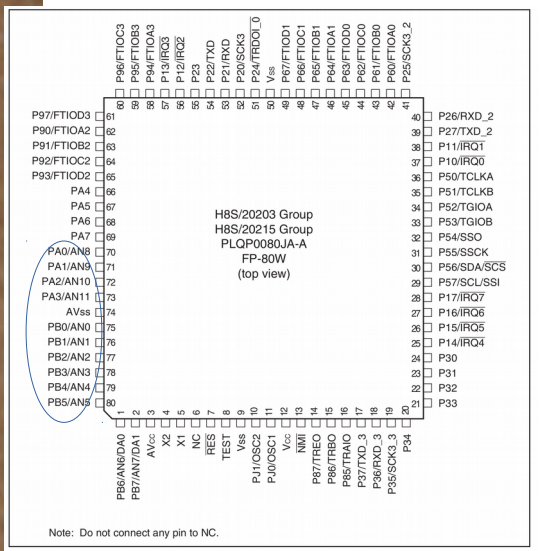
Privacy Issue

It is important to be able to detect energy theft, while still preserving the privacy of customers

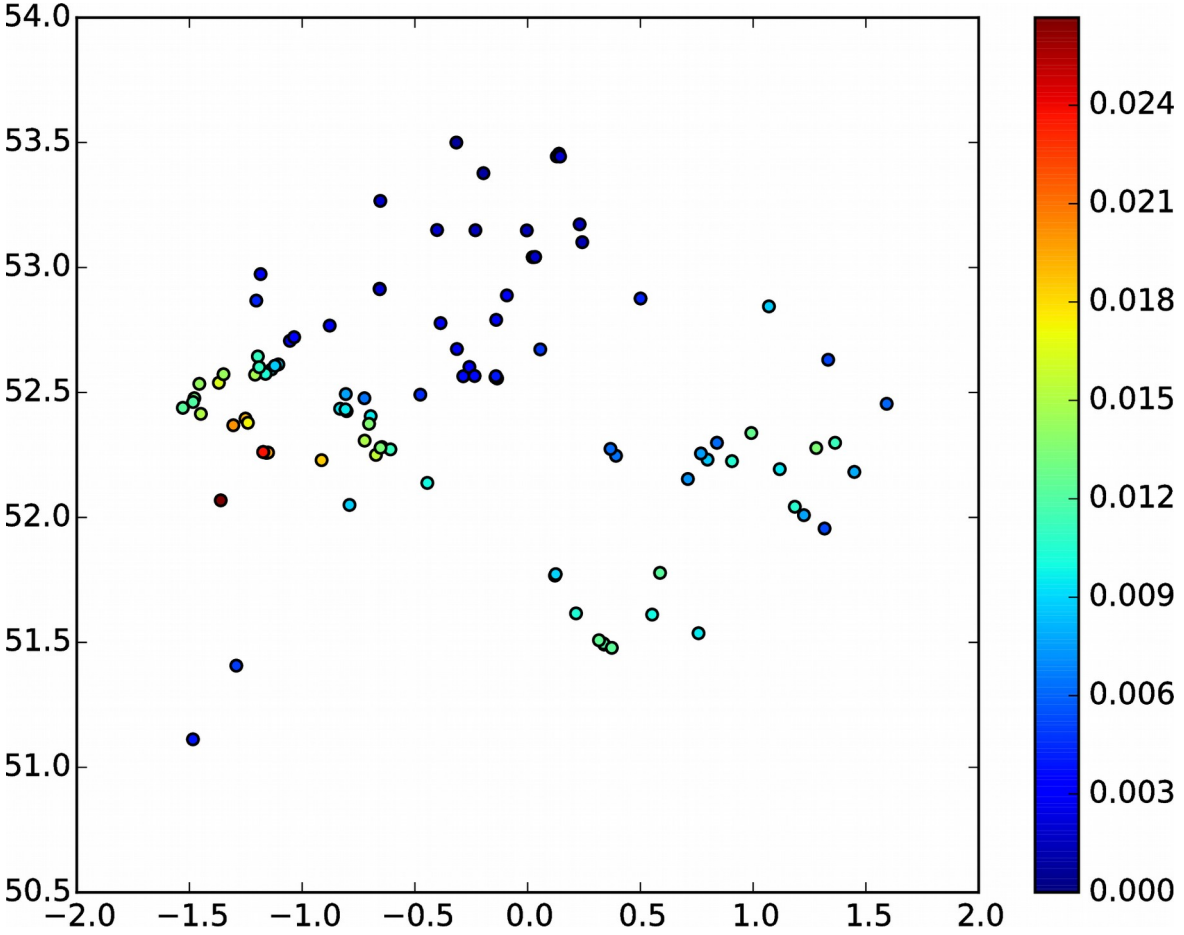
Fine grained data can leak sensitive information

Smart Meter Tampering



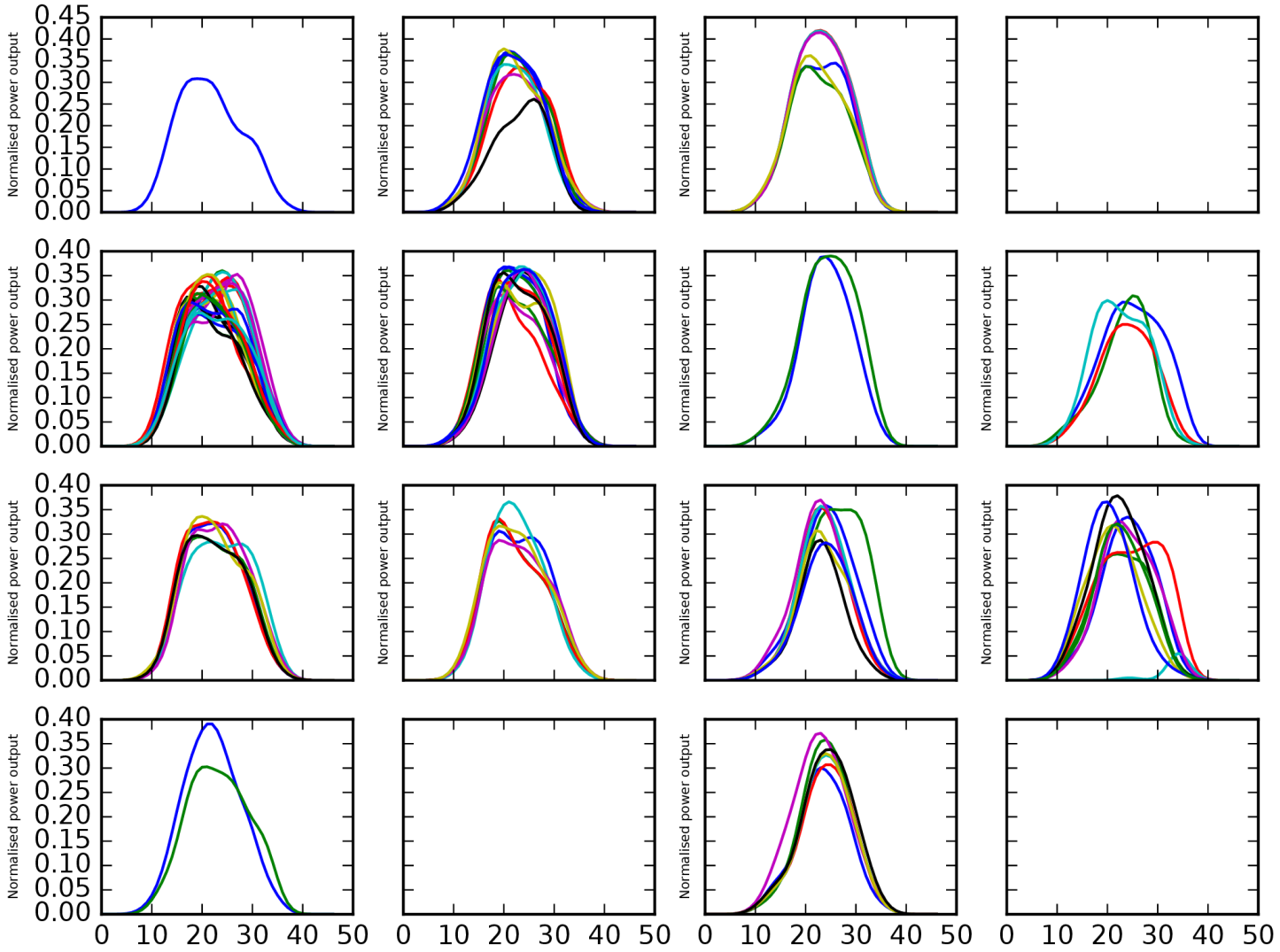


Distribution of solar nodes

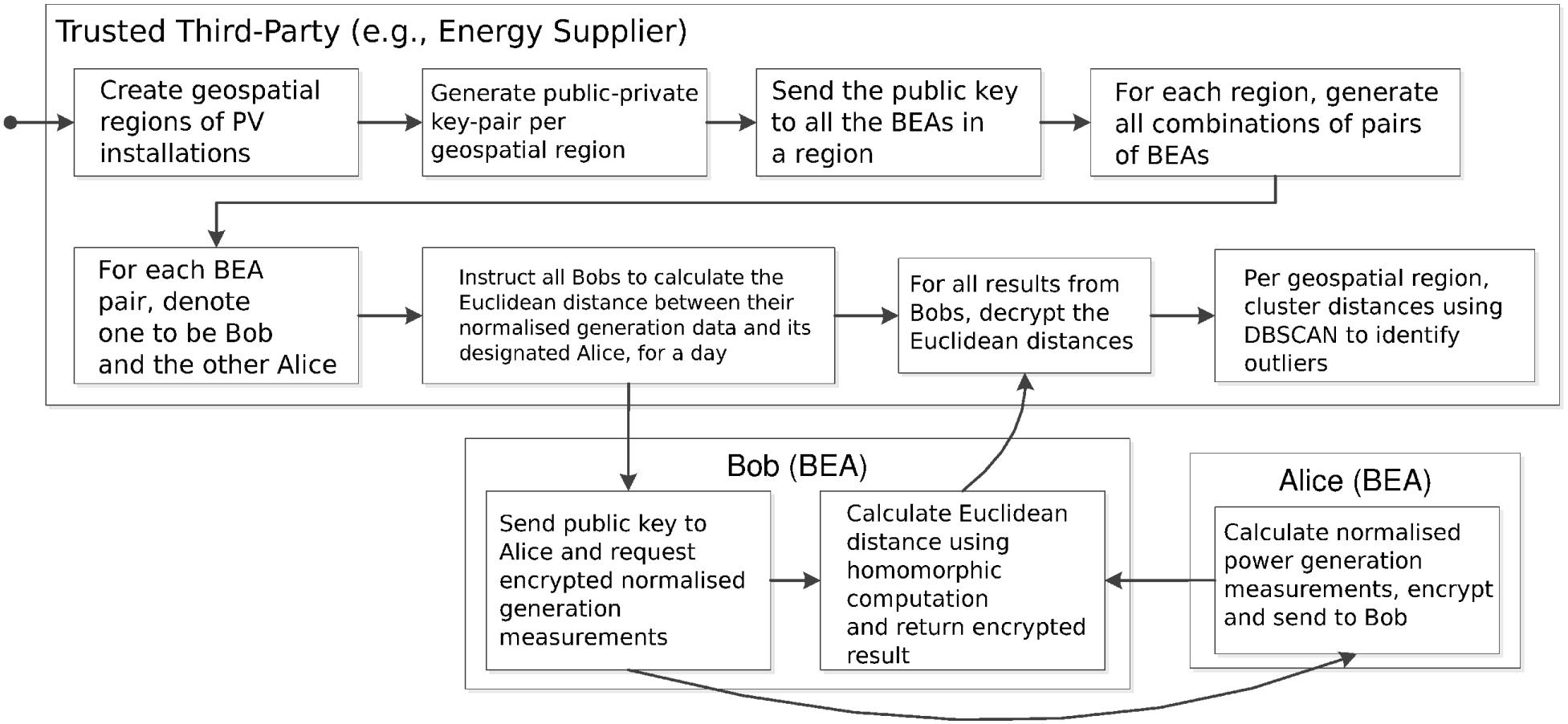


The data was obtained from the Sheffield Solar Group at the University of Sheffield: http://www.microgen-database.org.uk/

Normalised Solar Power Electricity Production



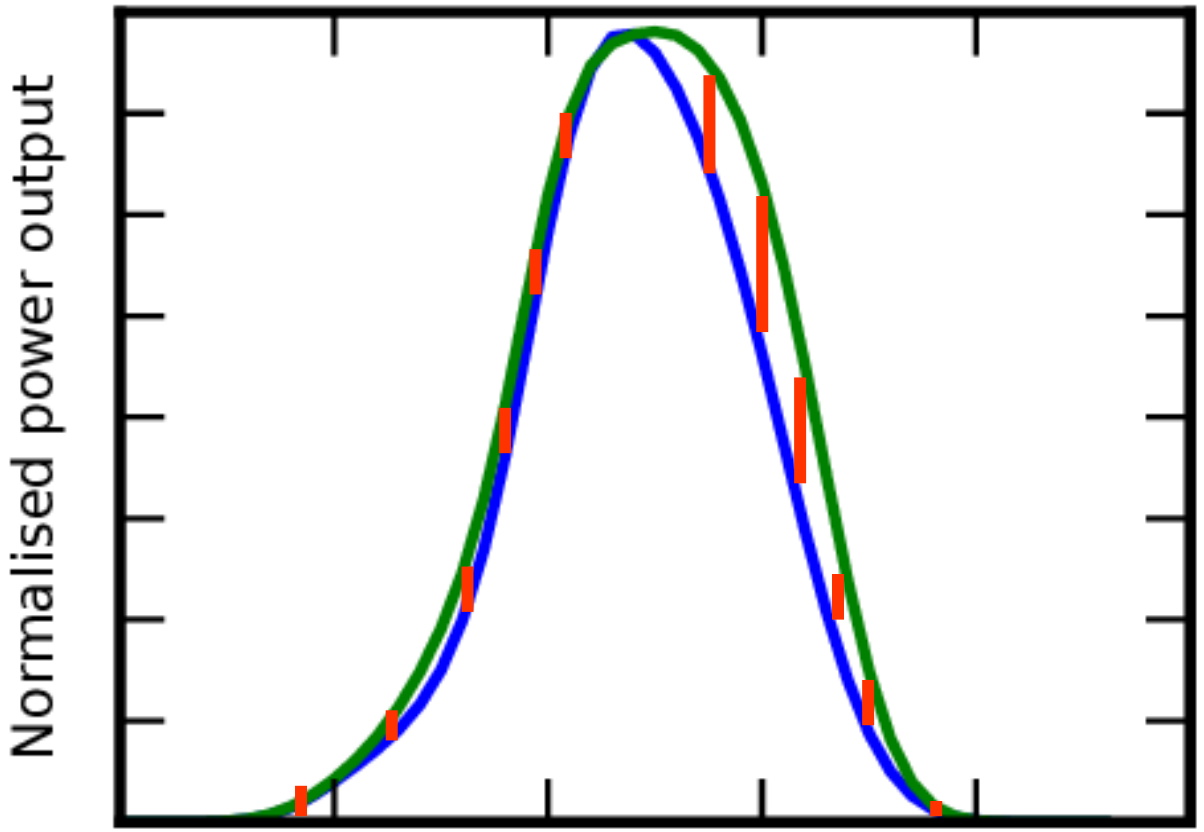
Process Diagram



Paillier Cryptosystem

- Partially homomorphic
- Supports addition operation
- Also supports multiplication, via use of a plaintext and ciphertext

Euclidean Distance



Sum of the difference between two houses solar panel outputs, across a day

Euclidean Distance

$$d(q_1, p_1) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Equivalent to the below (necessary as we can't perform exponentiation on ciphertexts where we don't know the plaintext with Paillier):

$$A = \sum_{i=1}^{n} q_i^2 \qquad B = \sum_{i=1}^{n} p_i^2 \qquad C = -\sum_{i=1}^{n} 2 q_i p_i$$

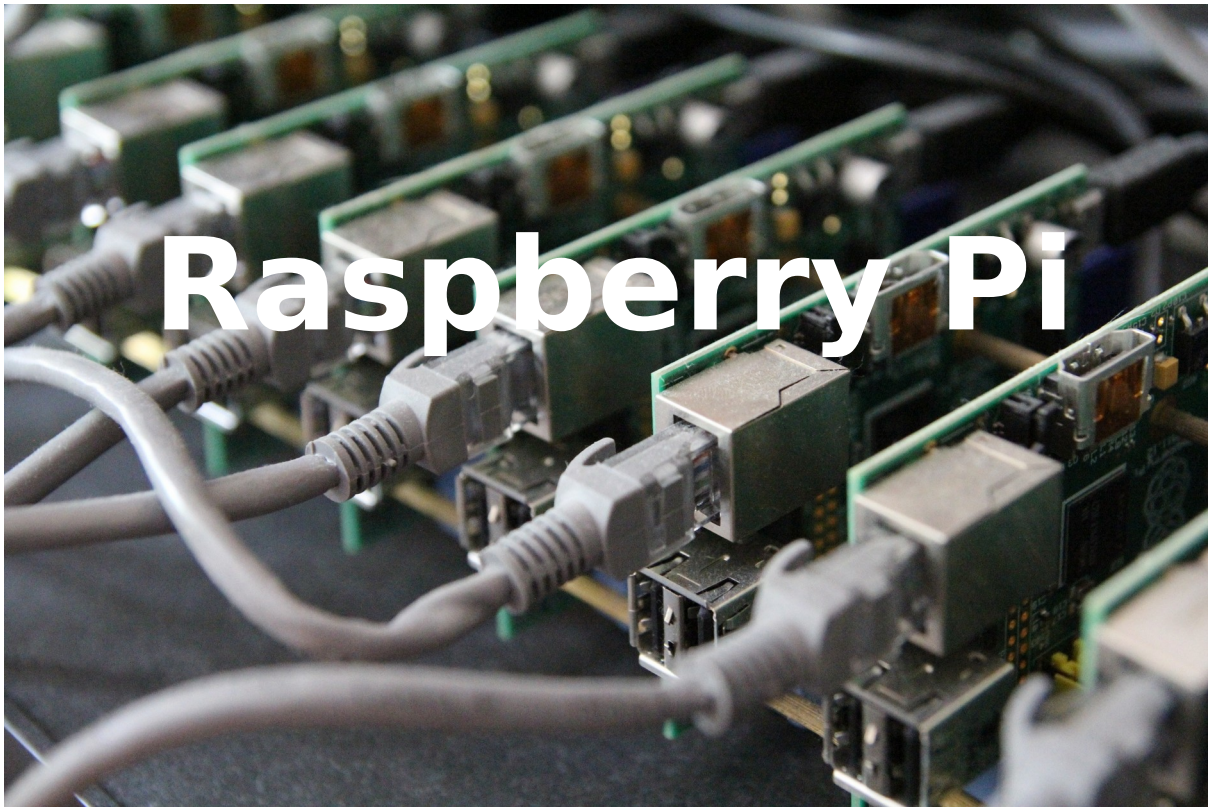
Plaintext Multiplied By ciphertext

$$d(q_1, p_1) = \sqrt{A + B + C}$$

See: S. D. Rane, W. Sun and A. Vetro, "Secure distortion computation among untrusting parties using homomorphic encryption," 2009

Experimental Platform

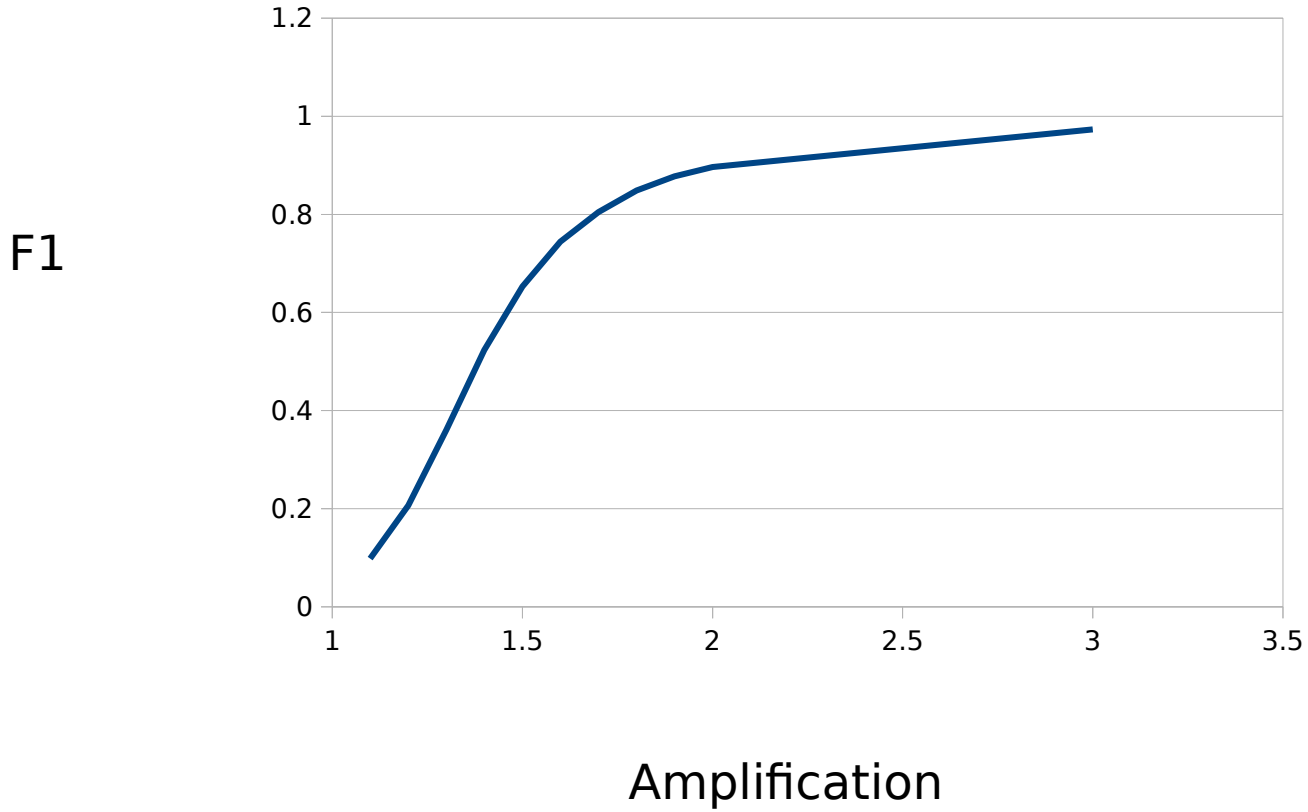
In ‘Software Architecture for a Smart Grids Test Facility - IT Implementation for an Emulated Low Voltage Smart Grid’ BEA is realised through the use an embedded industrial PC, such as the Siemens Nanobox PC SIMATIC IPC227D 19



& Mininet

Initial Results – F1 Score

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$



$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

Conclusion

- Privacy issues highlighted
- Preliminary tests of our system on two testbeds (Raspberry Pis & Mininet)
- Initial accuracy of results presented

Future Work

- Experimentation with differing geospatial sizes
- Filtering of data prior to euclidean distance
- Using similar techniques for attack detection in other systems

Any Questions?

Also feel free to e-mail me

at

c.richardson@lancaster.ac.uk