

# Engineering privacy-friendly computations

Dr George Danezis  
University College London.

# What is cryptography good for?

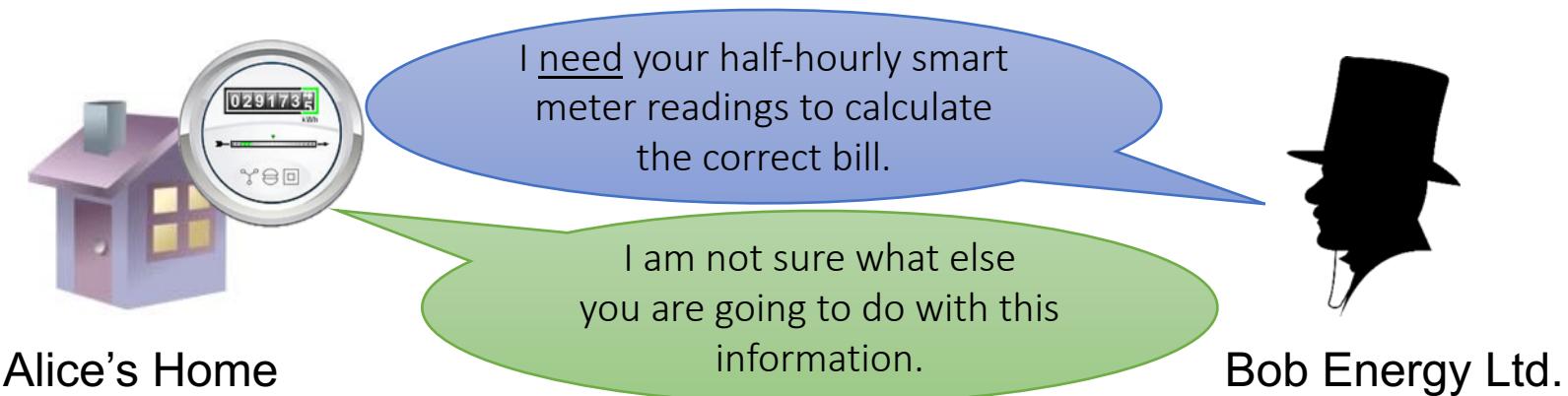


- Alice and Bob love each other ...

They can use TLS, IPsec, OTR, Tor to talk authentically and in secret.

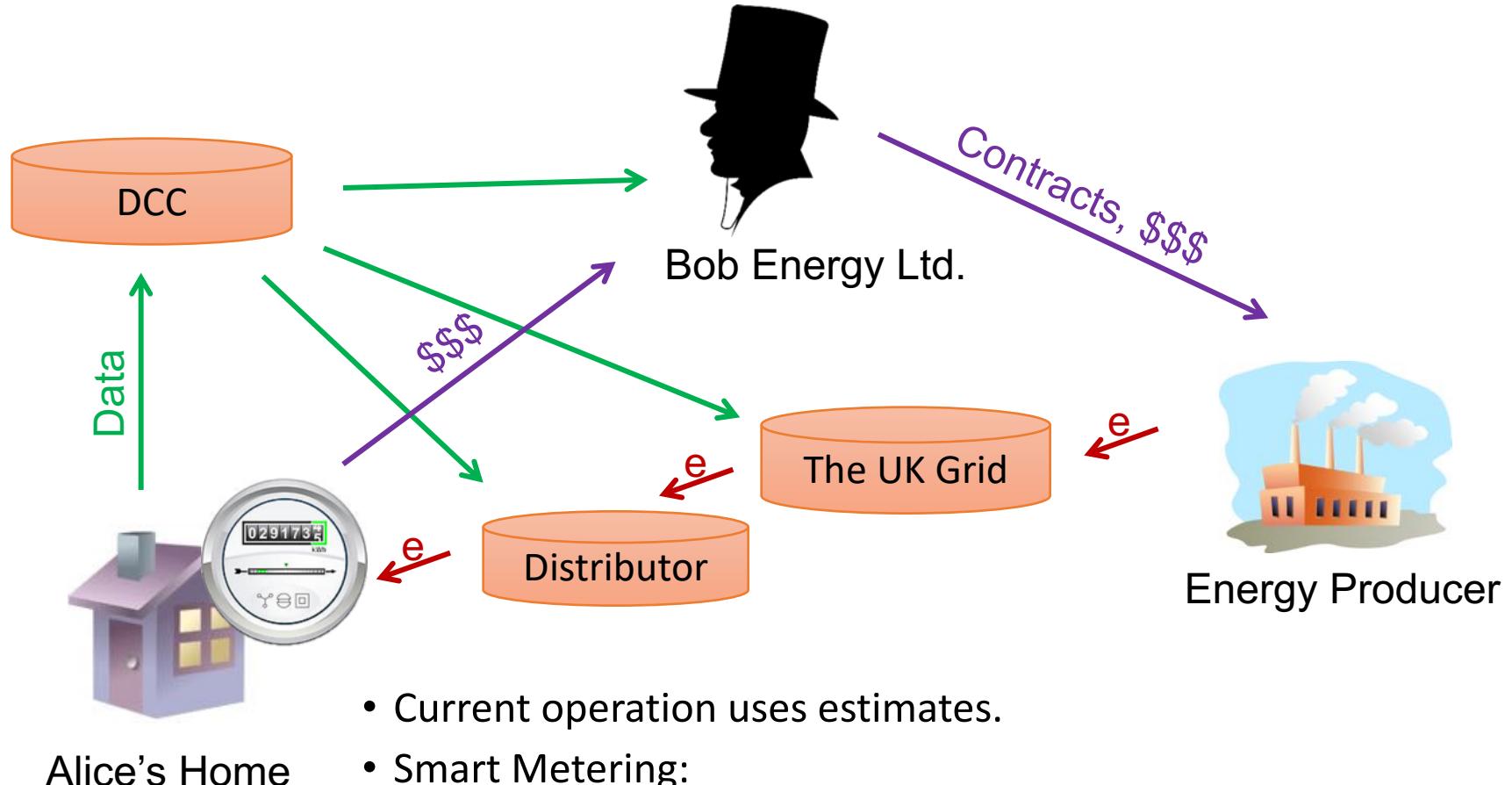
- The secure channel: a great success story for cryptography.
  - Except: deployed instances (TLS, IPsec, ...) are brittle.
  - Limits the imagination of the public and decision makers.
  - Does not solve the whole “privacy” problem.

# Alice and Bob do not trust each other



- The contemporary “privacy problem”:
  - Users, customers, citizens are asked to ...
  - ... share personal information...
  - ... with entities they would not trust to keep their secrets.
- Cannot be solved with a secure channel.
  - ... or signatures.

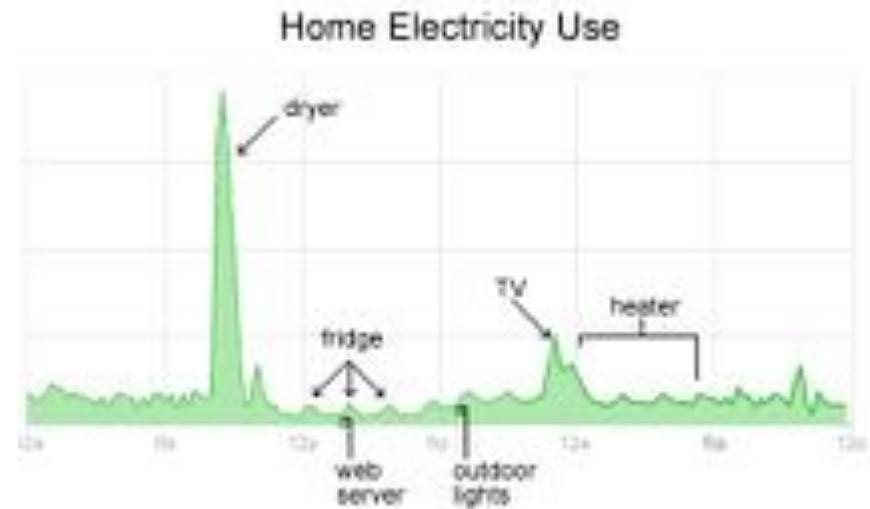
# A Smart electricity meter in every EU home by 2020 ...



- Current operation uses estimates.
- Smart Metering:
  - Record ½ hour readings and transmit them through a WAN.
  - Application: time-of-use bills, profiling.
  - Application: timely statistics, fraud detection.

# Are granular smart meter readings sensitive?

- Depending on the granularity:
  - Number of persons in the home.
  - Times when home empty.
  - Devices and appliances used.
  - Patterns of sleep.
  - Measures of wealth.
  - Mental and physical health.
- Data protection authorities recognize them as personal data.



# What can you learn from electricity? (I)

- Popular culture: Cannabis farms.
- Germany 1970s. Police used records of low energy consumption and cash based payment **to find the safe-house** of Rolf Heissler, a member of the Red Army Faction [5]. At the time this practice was ruled illegal.
- Hart [26]: **disaggregating appliances from consumption data**: First, a training phase and then an actual working phase. The training phase learns the electricity profile of different appliances, either by manually or automatically turning them on and off. The training can also be augmented or replaced by a database of known appliance signatures.
- Enev et al. [20] analyse electro-magnetic interference (EMI) of **television sets** on the power line and how it varies with the content displayed. They show that **movies can be identified** from EMI traces of 1200 known movies with up to 92% accuracy for some sets.

# What can you learn from electricity? (II)

- Greveler et al. [24] also describe an approach to **identify displayed TV channels**. They use the 0.5 hz smart meter measurements to create a prediction function that predicts the energy consumption of a dynamic back lighting LCD depending on the brightness of displayed content.
- Clark et al. [54] employ direct load monitoring at a computer with a sampling frequency of 1khz to identify **which website**, out of a pool of 8 popular websites, the computer is downloading and rendering.
- Lisovich et al. [40] predict inhabitants behaviour directly from smart metering data. They construct a sample disclosure metric that divides their behavior deductions into categories like **presence, sleep** schedule and others.

# Is there an alternative to sharing data?



## What are the benefits of me sharing my detailed energy data?

There may also be benefits to you sharing your detailed energy consumption data with your energy supplier or another supplier. For example:

- This information can be used to give you advice on how to save money on your energy bill and reduce your energy or carbon use
- It can give you more control over how much energy you're using
- It can be used to help you understand how much appliances are costing you and check if things are working properly
- Your energy supplier or a switching site could use the information to calculate if they can offer you a cheaper tariff
- Your energy supplier will need some information on how much energy you use (and when you use it in the case of time of use customers) to accurately bill you
- Tailored energy efficiency advice – based on accurate data specific to your home

- The subject of the rest of this talk!
- No. We do not “need” to share readings to:
  - compute time-of-use bills (or more).
  - extract aggregate statistics.

# A non-cryptographic privacy solution: Compute in the Smart Meter



Alice's Home

- Sensitive readings never leave the meter.
- Assurance:
  - tamper proof meter (same as for integrity of readings).
  - auditing outputs for privacy.



Bob Energy Ltd.

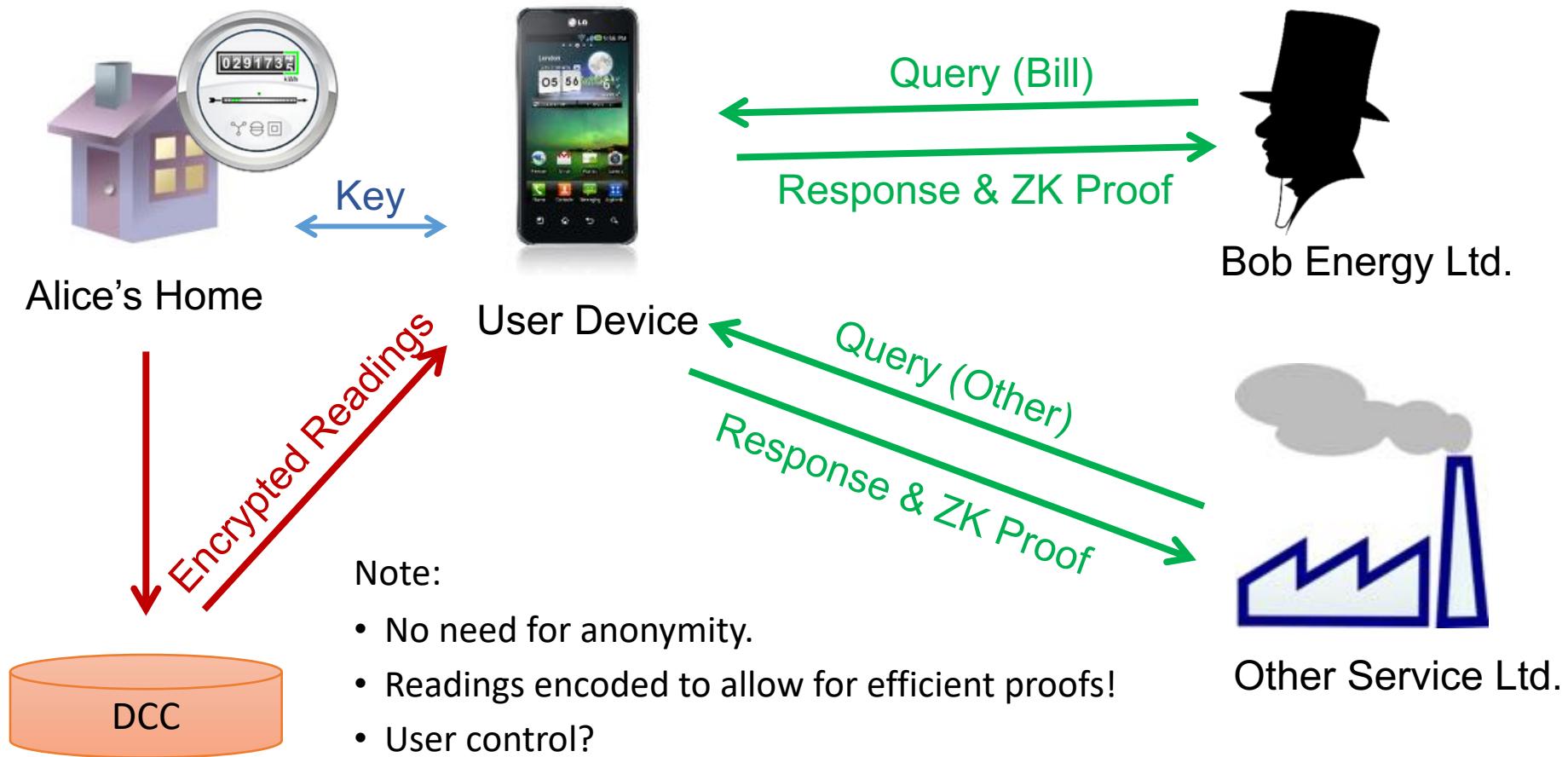
- Limitations:
  - What about other parties?
  - What about other computations? Aggregates?
  - Avoid mobile code, and generic interpreters.
  - Smart meters: they are not very powerful.

# Computational integrity: User centric private computation

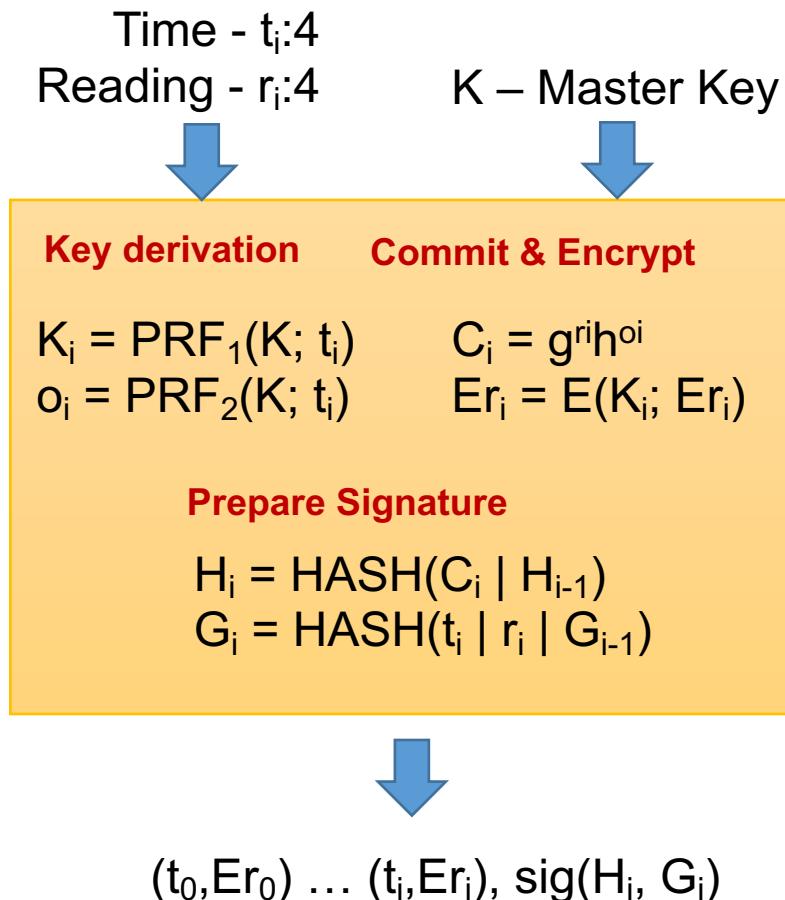


- Intuition:
  - Sensors are small, simple devices – no computation!
  - Users own a number of other computers.
  - Sensors record authoritative readings, computers compute, and send results to relying parties.
- Cryptography enables this!

# Smart metering & private billing



# Simple Meters



- Size considerations:
  - 4 + 4 bytes for each reading.
  - 2 hashes + signature per batch of readings.
  - Fits in the standard DLMS format.
  - No need to transmit commitments.
- Computation efficiency:
  - 2 PRF + 2 HASH + E
  - Can pre-compute most of the commitment + 32 mod-mult.
  - We can retrofit 8-bit microcontrollers to run it within the allowed time.
- Straight-line code, easier to verify.
  - Aizatulin, Gordon and Jürjens 2011

# Complex Computations

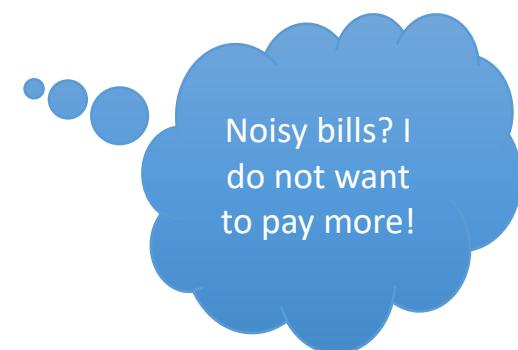
- How complex can billing be?
  - Simple: weighted sum of tariffs and readings – UK's bills in 12 days / 1 core.
  - Non-linear: tariff may change after a certain threshold.
  - Arbitrary: e.g. map each amount to an arbitrary bill.
- In general full zero-knowledge proofs of knowledge are required.
  - Key trick: how to prove a look-up in ZK? Use re-randomizable signatures (CL). (10Hz-100Hz)



- The power of Zero-knowledge lookups:
  - Prove that a substring is in a regular language in ZK.
  - Prove the classification result of a random forest classifier.
  - Important for generic user centric computations.
- The future: Pinocchio

# Results leak information

- What if the billing policy is crafted to leak information?
  - Trivial case:  $\sum w_i \times t_i$  with  $w_i = 1$  if  $t_i = T$  else 0 – leaks the value of  $r_T$
- Solution: add (positive) noise to the bill to mask readings.
  - Choose noise from a specific distribution.
  - Provide differential privacy guarantees.
- Oblivious accounting:
  - Each user keeps up-to-date tally of how much they overpay.
  - At the end of a long period (year) they can claim a rebate on their bills, to get back some of the overplayed amount.

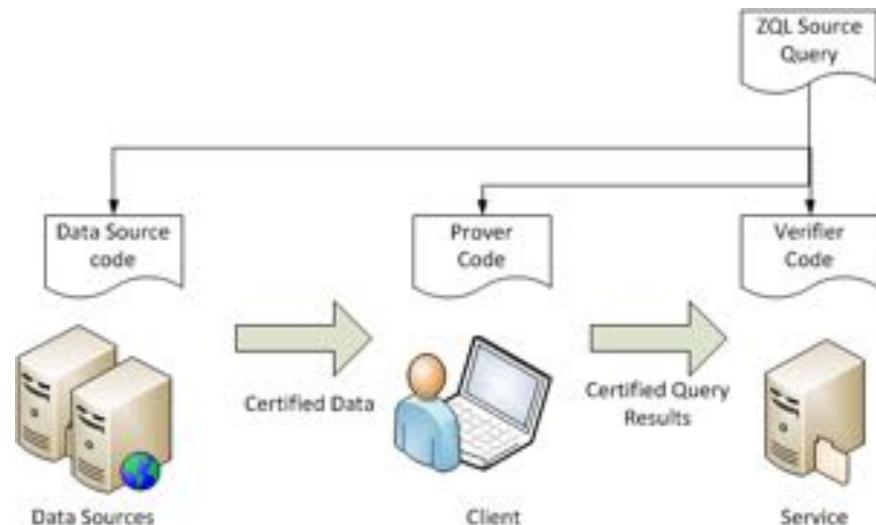


- Extreme privacy: Alice could prove that what she has paid in her account covers what she has consumed – with no more information about monthly bills.

# How to express computations?

- Fact: Engineers cannot write ZK proofs.
- ZQL: a language to express simple data processing.
  - No mention of cryptography.
  - Type annotations to denote public variables.
  - All other variables are private.
- Basic types and operators:
  - integers
  - tables
  - Arithmetic
  - Lookups
- Compile to:
  - code + prover.
  - Code + verification.

```
let smart_meter_bill
  ( $R$ : (int pub * int) table) // time, reading
  ( $T$ : (int * int) lookuptable) = // reading, fee
  ↓ (sum ((time, reading) → lookup reading  $T$ )  $R$ )
```



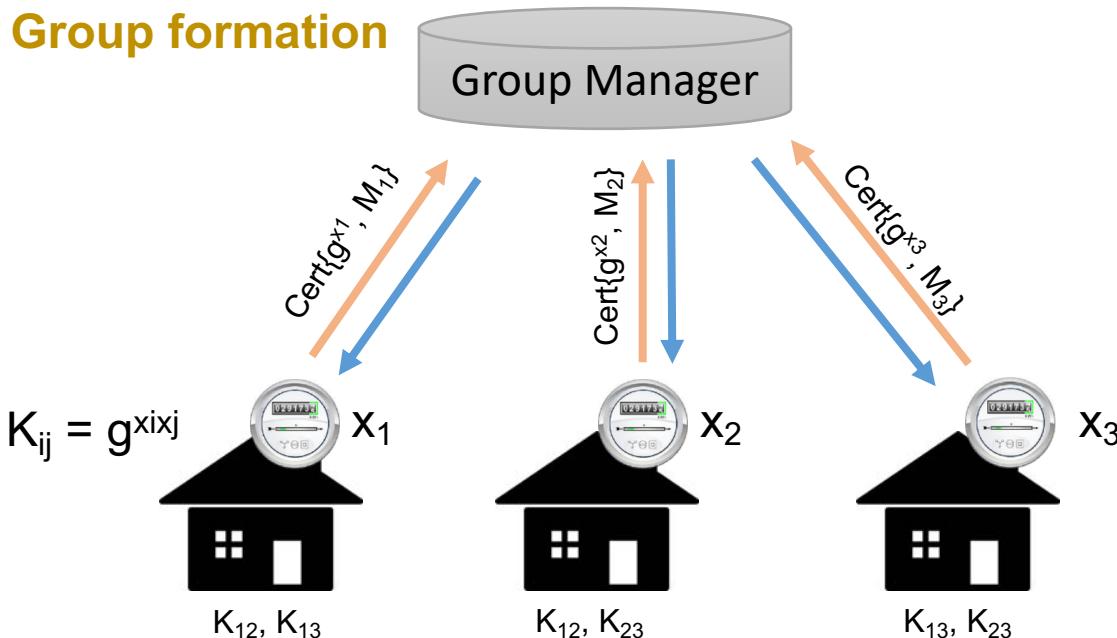
# Limitations of user centric computations

- What if we want to compute on **multiple users'** data?
- Privacy-friendly aggregates: e.g. simple sums
  - Day to day running of the grid: planning capacity for the next hours.
  - Distributors' network planning: ensuring peak capacity is sufficient.
  - Settlement process: do the contracts of a supplier cover customer base consumption?
- Theft detection: Is the sum of readings same as the aggregate meter?



# Secret sharing based aggregation

## Group formation



## Reading encryption

$$s_i = \sum_j (-1)^{j < i} PRF_{K_{ij}}(T_t)$$

$$c_i = E(r_i) = r_i + s_i \bmod 2^{32}$$

Note that:

$$\sum_i c_i = \sum_i r_i$$

- Two phases:
  - Group formation.
  - Reading Encryption.
- Group formation:
  - Key exchange
  - Managed.
  - Done once.
- Reading Encryption:
  - 4 byte ciphertext.
  - Only PRF used.
  - Compatible with previous scheme!

# Deployment and limitations



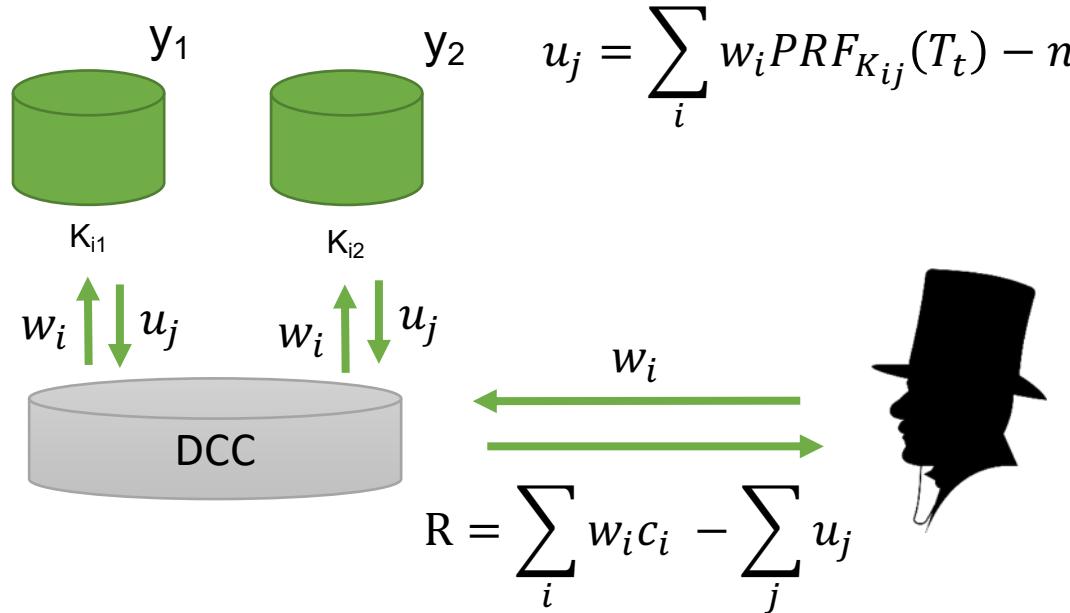
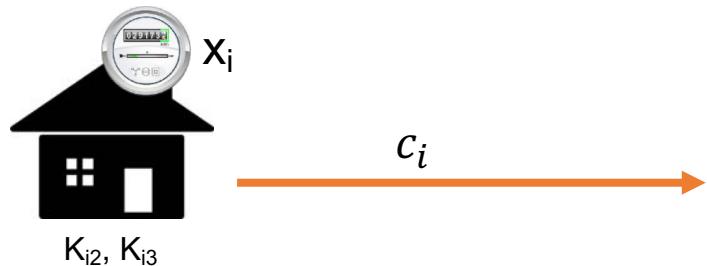
*Benessa Defend and Klaus Kursawe. Implementation of Privacy Friendly Aggregation for the Smart Grid. SEGS 2013.*

- ENCS is deploying the privacy-friendly aggregation protocols in a large test bed in the Netherlands.
  - With Aliander (DSO) and Elster (Smart meters)
  - Standardization effort.
- Limitations:
  - The size of the group is limited.
  - The more members the more storage for keys and computation to encrypt.
  - If a single meter in the group fails the aggregate is not available.
  - The groups are fixed – not so flexible.
  - Only simple sums are possible.

# Authority based aggregation

$$s_i = \sum_j PRF_{K_{ij}}(T_t)$$

$$c_i = E(r_i) = r_i + s_i \bmod 2^{32}$$

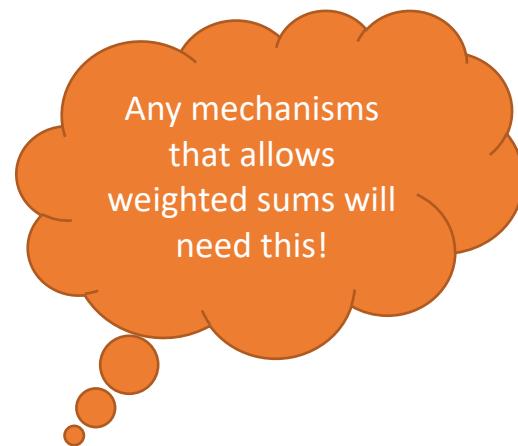


- Require: a number of authorities – one of them honest.
  - Meters: exactly the same functionality (by design).
  - Authorities: “unlock” result in a 1-round protocol with DCC.
  - Meter oblivious to further computations.

- Any weighted sum can be extracted: failing meters can be ignored.
- Differential Privacy through addition of noise ( $n$ ).
- Simple linear time-of-use billing protocol.
- Machine checked formal proof in easycrypt.

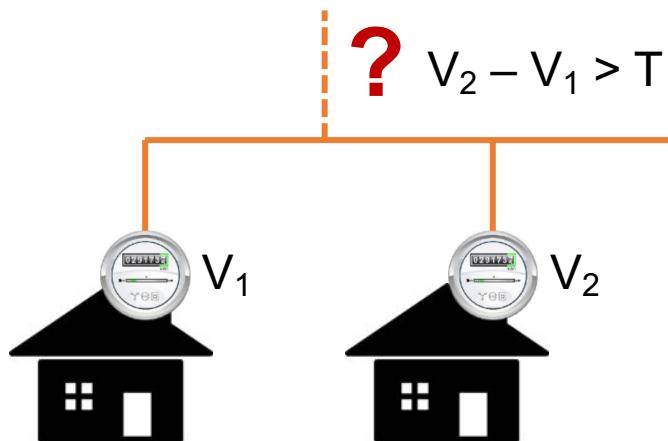
# The need for differential privacy

- Simple aggregation protocol:
  - fixed groups = fixed computations / time.
- Flexible aggregation protocol:
  - Arbitrary linear sums of secrets.
  - Attack:  $n$ -queries can extract  $n$  exact secrets.
- Solution:
  - Distributed differentially private mechanism.
  - Each authority adds some noise.
  - Downside: inaccurate results.
  - Option: some regulated computations are not noised.
  - Option: auditing model.



Any mechanisms  
that allows  
weighted sums will  
need this!

# Non-linear computations



---

```
def compare(c, rA, rB, Thld):
    diff = c.linear([1, -1], [rA, rB], -Thld)
    bits, _ = c.tobits(diff)
    return c.gneg(bits[-16])
```

---

## Line theft detection use-case

- Aggregation protocol: linear secret sharing based computation.
- Non-linear computations:
  - Use computations (mod p)
  - Same trick to only require 1 share per reading.
  - Authorities & DCC use SPDZ-like secret sharing computation.

# Key question: Whom to trust?

## User devices, meters, authorities

- User devices may be insecure.
- User devices may not always be available.
- The user may not have any devices.
  
- Meters are simple and insecure.
- Meters are under the control of suppliers (or others).
- Who knows what code, update run on a meter.
  
- What are the incentives to run an authority.
- What about them coming under compulsion to reveal keys.
- Worse for non-linear operations.
  
- One thing is sure:  
**Just giving the data to the supplier is the worse privacy option.**

# Key question: Which properties? Privacy, Integrity, public verifiability

- Current situation: no privacy. Integrity relies on meter and correctness of back end code.
- Ultimate integrity check: meters store 13 months of readings.
- Controversial: do we need integrity for the aggregation protocols?
  - Against malicious authorities?
- Verifiability for billing: currently pretty much none.
- In itself an advantage of the privacy friendly solutions.

# Key question: What is stopping deployment?

- Theory is cool: Zero-knowledge and secret sharing.
- Cost to implement protocols makes them prohibitive.
  - Cost = architecture changes & salaries of cryptographers.
- Generic protocols both a blessing and a curse
  - Can do “anything”.
  - Which means that it is hard to make them do “something”.
  - Not integrated in development tools, frameworks and libraries.
- The industry is still trying to transit to public key cryptography or elliptic curves.
  - Anything that is not security channels or certificates is not on the map.
  - New IT industries do not employ any cryptographers.
- Oh, and every law enforcement agency is happy enough to not see more privacy.
  - Data protection and consumer protection authorities do not employ cryptographers.

# All references

Alfredo Rial, George Danezis: **Privacy-preserving smart metering.** WPES 2011: 49-60

Klaus Kursawe, George Danezis, Markulf Kohlweiss: **Privacy-Friendly Aggregation for the Smart-Grid.** PETS 2011: 175-191

George Danezis, Markulf Kohlweiss, Alfredo Rial: **Differentially Private Billing with Rebates.** Information Hiding 2011: 148-162

George Danezis, Benjamin Livshits: **Towards ensuring client-side computational integrity.** CCSW 2011: 125-130

Andres Molina-Markham, George Danezis, Kevin Fu, Prashant J. Shenoy, David E. Irwin: **Designing Privacy-Preserving Smart Meters with Low-Cost Microcontrollers.** Financial Cryptography 2012: 239-253

Gilles Barthe, George Danezis, Benjamin Grégoire, César Kunz, Santiago Zanella Béguelin: **Verified Computational Differential Privacy with Applications to Smart Metering.** CSF 2013: 287-301

George Danezis, Cedric Fournet, Markulf Kohlweiss and Santiago Zanella-Béguelin. **Smart Meter Aggregation via Secret-Sharing.** ACM SEGS 2013: Smart Energy Grid Security Workshop, Berlin, 2013.

Carmela Troncoso, George Danezis, Eleni Kosta, Josep Balasch, Bart Preneel: **PriPAYD: Privacy-Friendly Pay-As-You-Drive Insurance.** IEEE Trans. Dependable Sec. Comput. 8(5): 742-755 (2011)

George Danezis, Markulf Kohlweiss, Benjamin Livshits, Alfredo Rial: **Private Client-Side Profiling with Random Forests and Hidden Markov Models.** Privacy Enhancing Technologies 2012: 18-37

Technology does not have to result in loss of privacy ...

**Engineering is all about options.**



**UCL ENGINEERING**  
Change the world