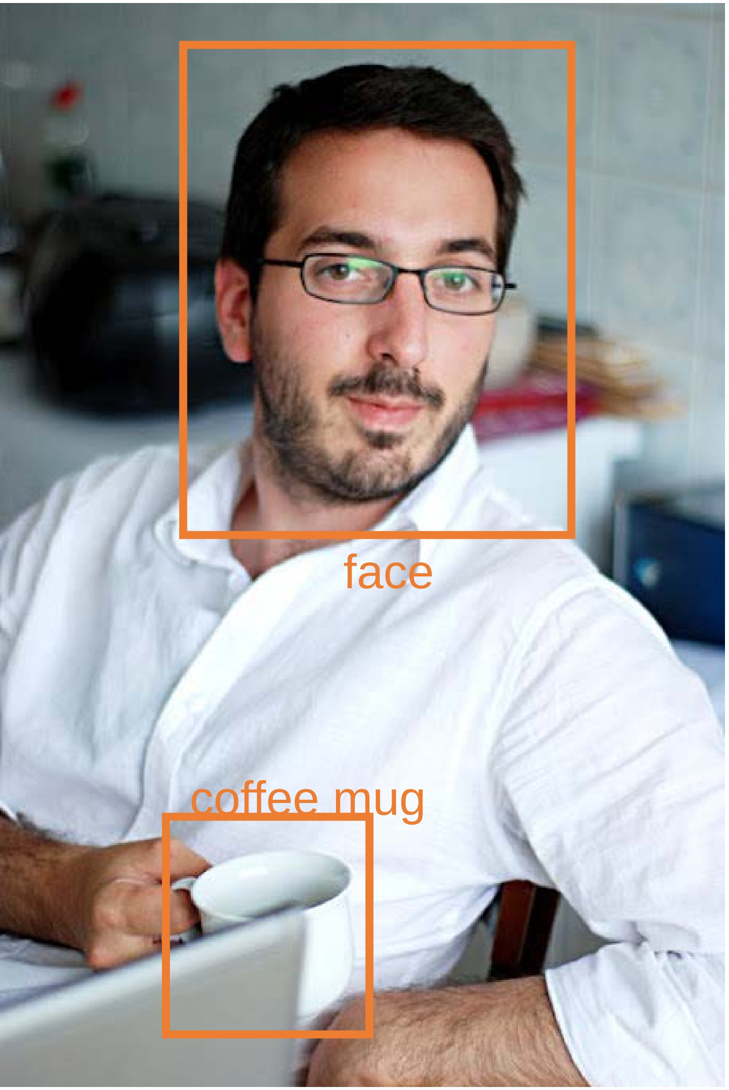
A Gentle Introduction to Privacy Enhancing Technologies & 2 Case Studies

Dr George Danezis
University College London

Privacy Summer School, 11 July 2016



Dr. George Danezis

- Short Bio:
  - PhD & Postdoc: Cambridge (2000-2005)
  - Visiting Scholar: KU Leuven (2005-2007)
  - Microsoft Research (2007-2013) – Researcher and Privacy Lead for Lab
  - University College London (2013) – Reader in Security and Privacy Engineering
- Expertise:
  - **Privacy technologies**:
    anonymous communications, traffic analysis, location privacy, private computations, applied cryptography, …
  - **Peer-to-peer & networking security:**
    Sybil attacks / defences, secure DHCP, private presence, …
  - **Social networking security:**
    Automatic privacy policies, community detection, …
  - **Machine learning and security**:
    Private classification, Bayesian inference, de-anonymization, …
  - **Infrastructure and services security:**
    Smart metering security, pay-as-you-drive insurance, electronic cash, …

Outline

Core:

- Introduction to Privacy Technologies
- Case Study I: Pay as you drive insurance & privacy
- Case Study II: Privacy technologies for smart grid

Cryptography & Privacy Enhancing Technologies

A gentle introduction

PETs & their "threat models"

- Cryptography is used to build technologies that protect privacy.
  - Traditional: Confidentiality, control, or even information self-determination.
  - Privacy a bit different than traditional confidentiality.

- What makes Privacy Enhancing Technologies (PETs) different:
  - Threat model: weak actors, powerful adversaries.
  - Susceptibility to compulsion.
  - Cannot assume the existence of Trusted Third Parties (TTP):
  - 5Cs: Cost, Collusion, Compulsion, Corruption, Carelessness.

- PETs design principles:
  - Rely on end-user devices. (Challenge here!)
  - Distribute trust across multiple semi-trusted third parties.
  - Allow users to chose who they trust for certain operations.
  - Use cryptography to ensure confidentiality and correctness.
  - Keep only short term secrets, if any.

Perfect Forward Secrecy

- Encryption can be used to keep communications secret.
  - But what if someone forces you to disclose the key?

- Perfect Forward Secrecy (PFS): gold standard of encrypted communications.
  - Start with keys that allow Alice to authenticate Bob.
  - Alice and Bob create fresh public keys and exchange them.
  - They establish fresh shared keys, and talk secretly.
  - Once done, they delete the shared keys.

- Result: after a conversation is over, no-one can decrypt what was said.
  - Additional property: plausible deniability.
  - Illustrates: using only end devices, no long-term keys.

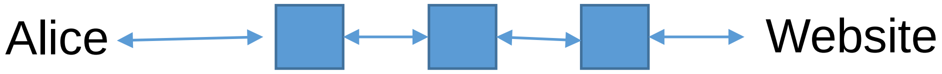
- Available now: Off-the-record (OTR), Signal (Android / iOS).
  - Download "Signal" and use it! Red Phone encrypts calls.

Protecting communications meta-data

- Who talks with whom, and what you browse is sensitive.
  - Alice talks to Bob, Bob is a cancer doctor.
  - Alice Browses the NHS website, looking at pages on STDs.

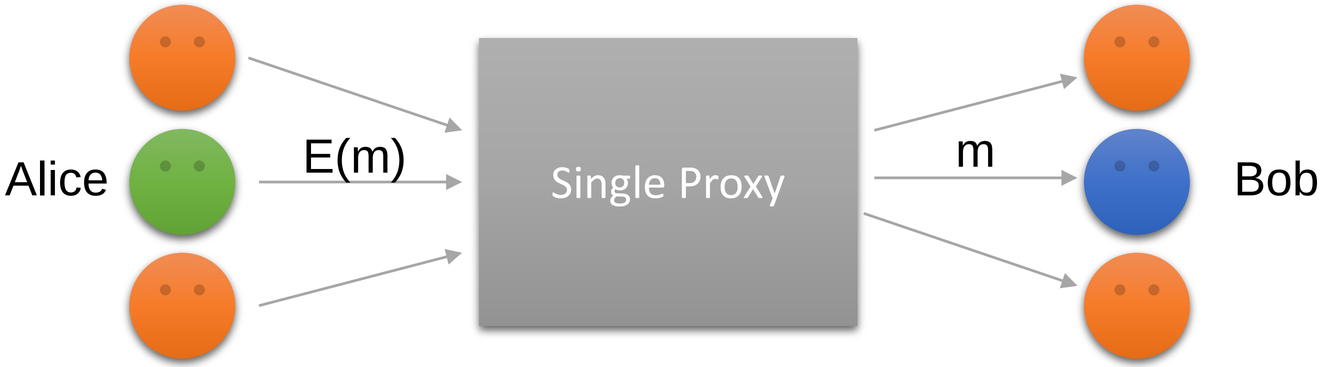
- Extensive research shows a lot can be inferred from meta-data:
  - Sender, receiver, length & time of communication, pattern.
  - Eg. mental condition, personality, language, emotions, political opinion.
  - Even if the content is encrypted!

- Anonymous communication systems hide such information:
  - Best known: Tor – The Onion Router.
  - How? Use a set of relays:



  - Illustrates: distribute trust, chose who to trust, crypto …

Proxies for Anonymous Communications

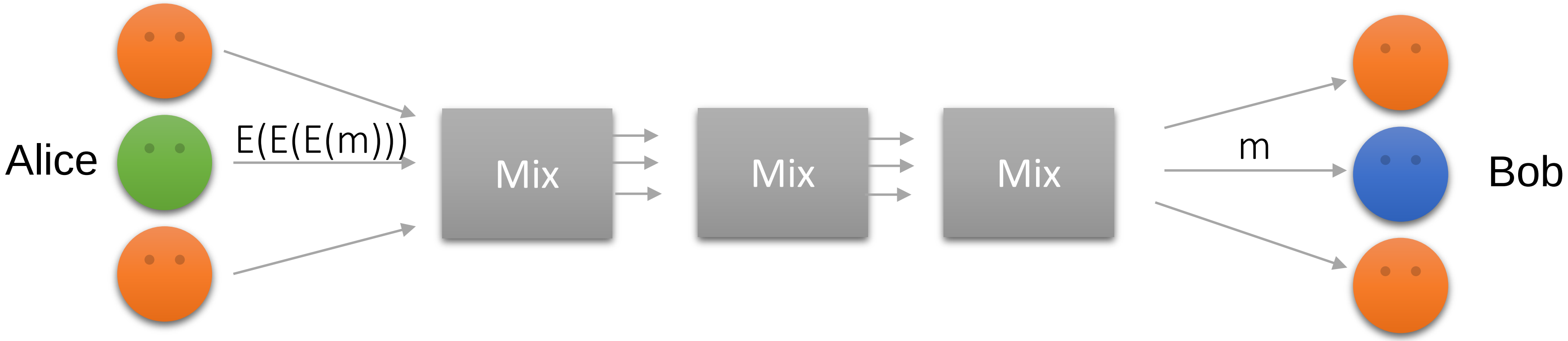


- Alice wants to hide the fact she is sending a message to Bob.
  - The proxy decrypts the message.
  - The proxy batches many messages.
  - The proxy is the TCB.

- Problem:
  - Low throughput.
  - Corrupt Proxy or Proxy hacked / coerced.
  - Real case: Penet.fi vs the church of scientology (1996)

Danezis, George, Claudia Diaz, and Paul Syverson. "**Systems for anonymous communication.**"
Handbook of Financial Cryptography and Security, Cryptography and Network Security Series (2009): 341-389.

Mix Networks and Onion Routing



- Solution: Use multiple cryptographic relays (mix)
  - Sender encrypts messages using multiple keys, through a set of mixes.
  - Each mix batches multiple messages.
  - TCB: Not a single mix, or client. No single place to coerce to trace everyone.

- From mix-networks to Onion Routing
  - OR: sender sends a stream of messages through the sequence of relays.
  - Problem: timing of traffic leads to correlation (c^2 attack)
  - Distributed TCB: adversary can compromise some circuits not all.

Hayes, Jamie, and George Danezis. "**Better open-world website fingerprinting**." *arXiv preprint arXiv:1509.00789* (2015).
Murdoch, Steven J., and George Danezis. "**Low-cost traffic analysis of Tor**." Security and Privacy, 2005 IEEE Symposium.

Private Information Retrieval

- Key problem: which database record you access is sensitive!
  - Example: which book you are looking at the library?
    Which friend you check if they are on-line?
    What music you are listening?
    Which minister you look up in your online address book?

- PETs Solutions:
  - Private information retrieval: access a public record without leaking which – even to the provider! (Is that even possible?)
  - ORAM: access your own private encrypted records, without divulging which (cheap) to cloud store.

- Techniques: distribute trust, rely on client (e2e).

Private Computations in general

- Alice and Bob want to work out who is older, without telling each other their age – can they do that?
  - Amazing result: any function that could be privately computed by providing the private inputs to a trusted third party, can also be computed privately.
  - Ie. Alice and Bob simply exchange cryptographic messages, and end up with the result! Neither of them learns the other’s age!
  - Also enables secure outsourcing.

- Two families of techniques:
  - Secure Multiparty Computation: well established and understood techniques based on secret sharing data.
    Commercial support (eg. Cybernetica’s Sharemind).
  - Homomorphic Encryption: allows operations on encrypted data.
    Toy Prototypes.

- Warning: slow for generic computations.
  - Normal CPU 1,000,000,000s (GHz) of operations a second.
  - Secure computations 1-10 per second (Hz) in general.

Specific Private Computations

- Generic Private Computations slow – but specific ones can be fast.
  - Smart metering examples: aggregation, fraud detection, billing.
  - Private road tolls.
  - Private authentication and authorization.
  - Simple private statistics.
  - Detecting common elements in sets.

- Application specific protocols can be practical.
  - But they need to be evaluated, and the computation needs to be simple.
  - High-value simple computations are commonplace.

- Example deployments: ENCS test-best deployment of privacy-friendly aggregation for smart metering / smart grid roll-outs.

Zero-knowledge Proofs

- PETs: 10% confidentiality, 90% making sure no one cheats.
  - Key: protect users from each other.
  - Key: protect users from corrupt elements of the infrastructure.
- The challenge: need to prove that something was done correctly, without revealing and private information.
  - Eg. the electricity bill was computed correctly, but hide how much electricity was consumed at specific times.
  - Eg. I am old enough to have a drink, but I will not tell you my age.
- "Zero-knowledge" proofs – allow you to prove statements about secret values, without revealing them.

How mature are PETs?

Maturity

- Not all PETs are equally well understood and mature for use.
  - PFS: download “Signal” now. Demand it everywhere. 1B users (Whatsapp).
  - Anonymity: Tor provides a weak form of anonymity, 1M users.
  - ZKP: Pilots (EU Prime, Primelife, ABC4Trust)
  - Specific Private Computations: pilots (Tor statistics & ENCS smart metering)
  - PIR / ORAM: we can build it, not large scale deployments.
  - Generic Private Computations: start-ups & pilots (Cybernetica & Microsoft)

Performance

- Performance:
  - Encryption of communications and storage: super-fast, will not slow down anything you care about.
  - ZKP: slow, but usually need to prove simple things.
  - Anonymity / PIR / ORAM: is slower than normal communications.
  - Private Computations: much slower – 6-9 orders of magnitude.

Privacy Beyond Cryptography

Anonymization, controls on usage, and logging

Other ways to protect privacy

- Non-cryptographic technologies are also used to protect privacy.
- They have their uses, particularly where a trusted third party exists.
  - Remember the 5Cs: cost, compulsion, collusion, corruption, carelessness.
- However some mechanisms are misunderstood:
  - Dataset anonymization.
  - Query Privacy / Privacy in statistical databases.
  - Restricting use of collected data.
  - Logging to detect privacy compromises.

Data Anonymization

- “Would it not be nice if: you can take a dataset full of private data, and transform it into one with no private data – while keeping all the value of the data?”
  - Magical thinking: this cannot happen in general.

- The problem of de-anonymization:
  - Any aspect of the “anonymized” dataset can be used to link the records to known named records.
  - Example of Netflix (anonymous) vs. DBLP (named) de-anonymization.
  - In general it is impossible to sanitise only the private information without severely scrubbing all the usefulness out of the dataset.
  - Removing PII is not enough!

- Data anonymization is a weak privacy mechanism. Only to be used when other (contractual, organizational) protections are also applied.

Query Privacy

- “Would it not be nice if I could send complex queries to a database to extract statistics, and it returned results that are informative, but leak very little information about any individual?”
  - Possible: state of the art are “differential privacy” mechanisms.

- Why is that possible (while anonymization was impossible):
  - The final result depends on multiple personal records.
  - However it does not depend much on any particular one (sensitivity).
  - Therefore adding a little bit of noise to the result, suffices to hide any record contribution.
  - In the case of anonymization: need to add a lot of noise to all the entries.

- Example: average height in the room via anonymization or query privacy.

- Public policy:
  - Notice the difference in the share of the architecture to provide robust privacy.
  - Notice that a TTP holds the data.

Controls on usage of collected data

- “Limiting collection is not practical, so why not place stricter limits on use instead?” - Favourite of Craig Mundie (ex-Microsoft)

- In practice: use some access control mechanism to ensure that once collected the data in only used for some things.

- Problems of this approach:
  - How does the user, or anyone else, gets assurance of robustness?
  - Abuses are invisible making this more difficult to police.
  - Technically need to keep track of private data and policies – even more complex than ensuring it is not collected.
  - Need to ensure consent for use, even more difficult than consent for collected (since user may not even be available – bulk datasets).

- Nearly no research on how to robustly achieve this, and prevent abuse.
  - Basically: “trust us we would never do anything wrong”.
  - No clear direction to design such robust technologies.

A cautionary note on more logs

- “Well it is simple: you collect all the data, and then you audit all operations and access to it. Thus if anyone abuses it you can find them and punish them”
    - So many problems with this …

- Issues:
    - Authorized accesses are themselves sensitive: eg. accesses to medical records. Access to contacts.
    - It is not guaranteed that the unauthorized access was not itself the result of a compromised user in the organization.
    - Once personal data leaks it cannot be put back in the bottle. Eg. the leakage of private pictures of celebrities.

- Public Policy: detecting compromises after the fact is one of the weakest security mechanism, and a weak privacy mechanism. It is not even clear someone can get punished.

Public verifiability and protection

- “How do I know this software I am using provides a gold standard level of privacy protection through PETs?”
  - Key question!
- Answer 1: we leave it up to everyone to examine!
  - Enormous externality – each user must be an expert and check.
- Answer 2: provide clear specifications, require applications providing privacy to provide transparency in their code & operations.
  - Concept of “Public verifiability” of both code and operations.
  - Gold Standard in the world of PETs (PGP, Tor, Signal, …)
  - Reluctance from industries to adopt for PETs or anything else.
  - Serious public policy issue beyond PETs (VW scandal).
- At what point does society have a right to know how key machinery works?
  - Remember: this was the aim of patents, originally.
  - This space will require serious regulation & consumer protection.

In conclusion

- Cryptography is everywhere, mostly as a key tool to secure telecommunications and transactions – not privacy.
- Cryptographic primitives can be used to build PETs.
  - Some of those are very mature (encryption, anonymity systems), and know-how on how to build them more and more commonplace.
  - Some are less mature (private computations, PIR), but still useful in special cases.
- Some common non-cryptographic privacy protections need careful thought.
  - Dataset anonymization and logging are weak at best.
  - Query privacy can be robust, given the correct architecture.

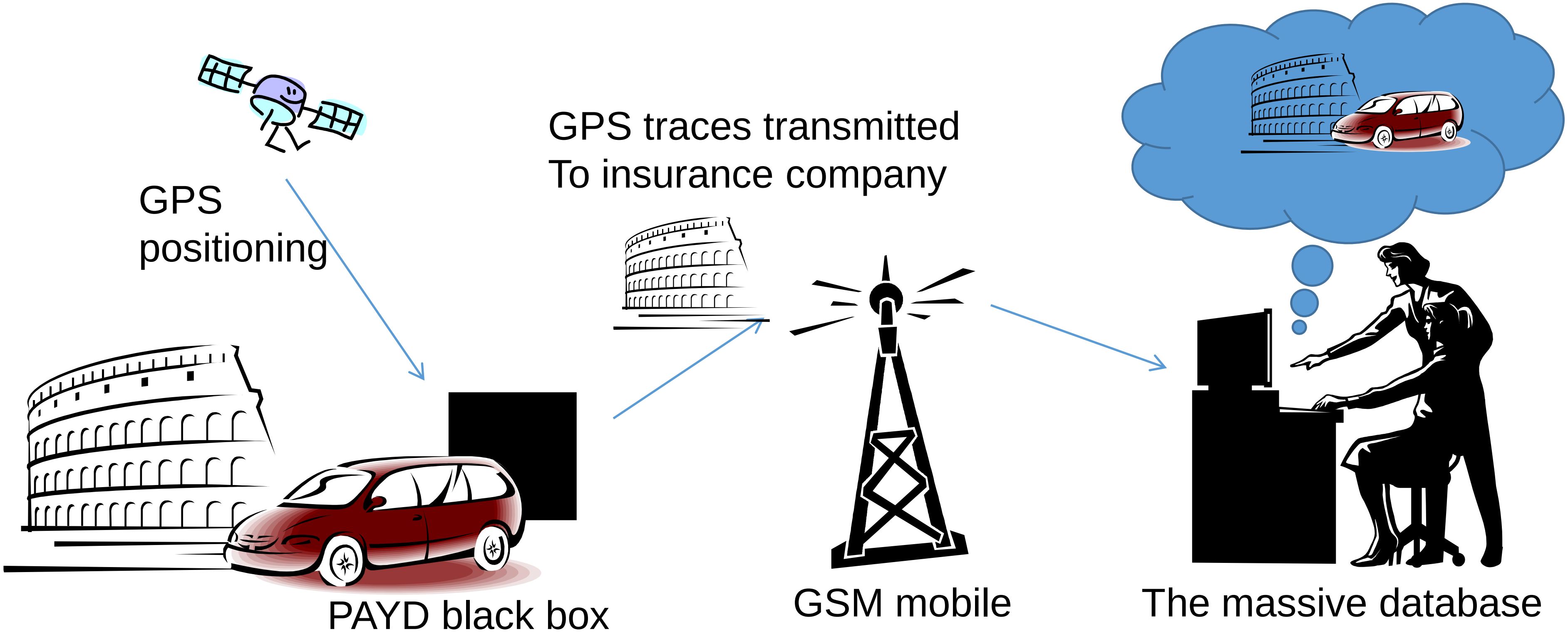
Case study:
Pay as you drive insurance

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)

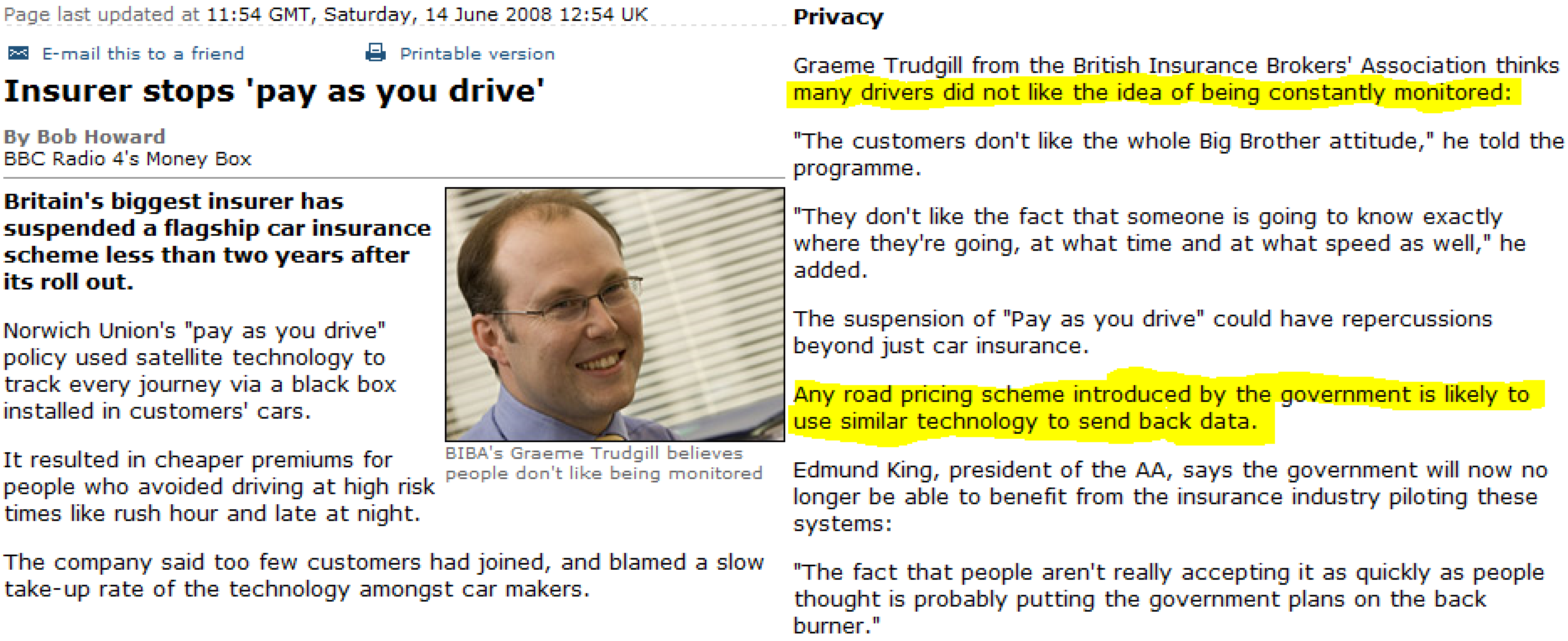
Josep Balasch, Alfredo Rial, Carmela Troncoso, Bart Preneel, Ingrid Verbauwhede, Christophe Geuens: PrETP: Privacy-Preserving Electronic Toll Pricing. USENIX Security Symposium 2010: 63-78

A case study: PAYD pilot

- PAYD: Pay as you drive insurance model
  - Aim: reduce driving on risk roads and times & cost



PAYD Fail!



Page last updated at 11:54 GMT, Saturday, 14 June 2008 12:54 UK

E-mail this to a friend   Printable version

Insurer stops 'pay as you drive'

By Bob Howard
BBC Radio 4's Money Box

Britain's biggest insurer has suspended a flagship car insurance scheme less than two years after its roll out.

BIBA's Graeme Trudgill believes people don't like being monitored

Norwich Union's "pay as you drive" policy used satellite technology to track every journey via a black box installed in customers' cars.

It resulted in cheaper premiums for people who avoided driving at high risk times like rush hour and late at night.

The company said too few customers had joined, and blamed a slow take-up rate of the technology amongst car makers.

Privacy

Graeme Trudgill from the British Insurance Brokers' Association thinks many drivers did not like the idea of being constantly monitored:

"The customers don't like the whole Big Brother attitude," he told the programme.

"They don't like the fact that someone is going to know exactly where they're going, at what time and at what speed as well," he added.

The suspension of "Pay as you drive" could have repercussions beyond just car insurance.

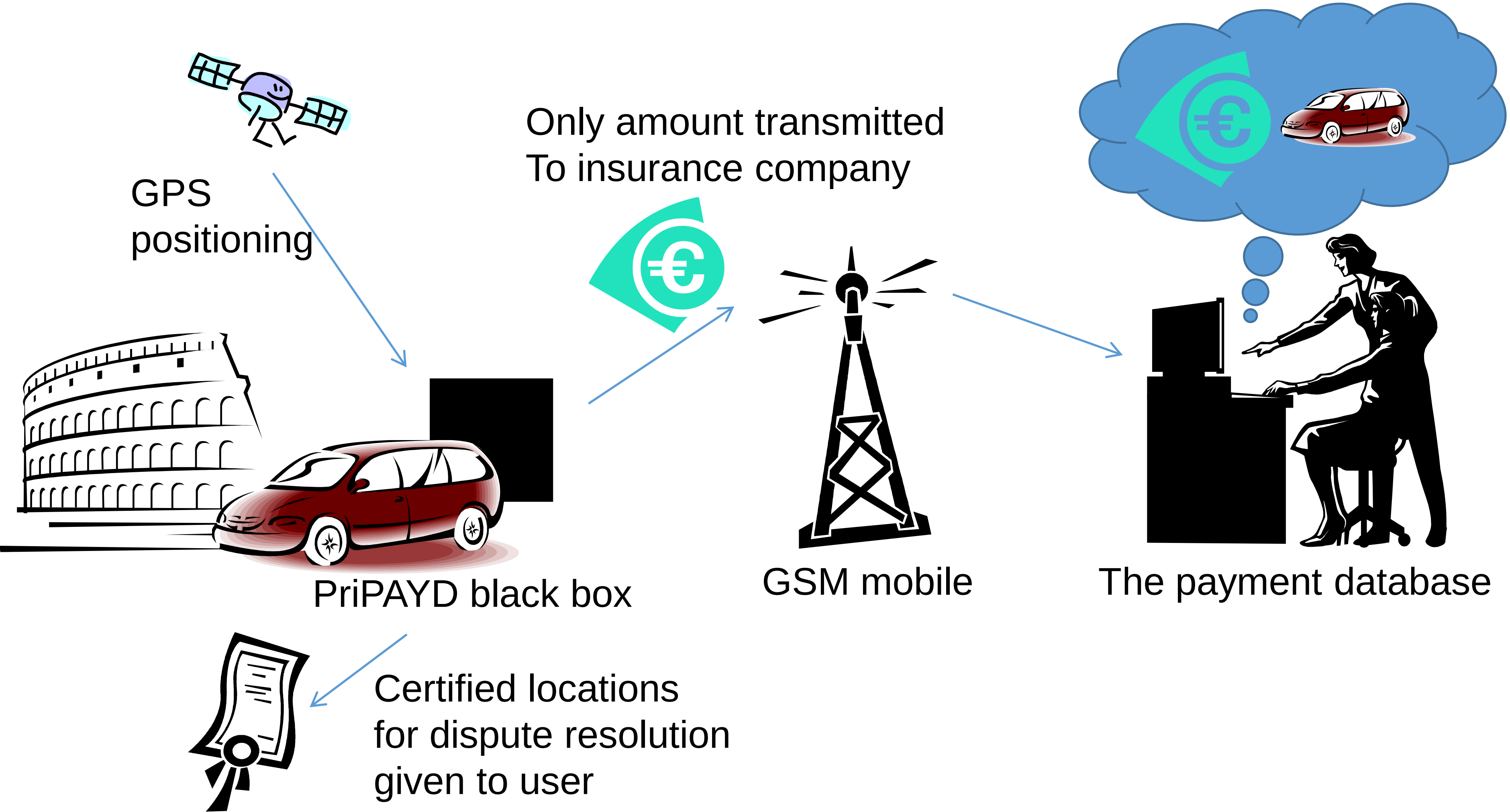
Any road pricing scheme introduced by the government is likely to use similar technology to send back data.

Edmund King, president of the AA, says the government will now no longer be able to benefit from the insurance industry piloting these systems:

"The fact that people aren't really accepting it as quickly as people thought is probably putting the government plans on the back burner."

PriPAYD

- Privacy friendly PAYD design



GPS positioning

Only amount transmitted
To insurance company

PriPAYD black box

GSM mobile

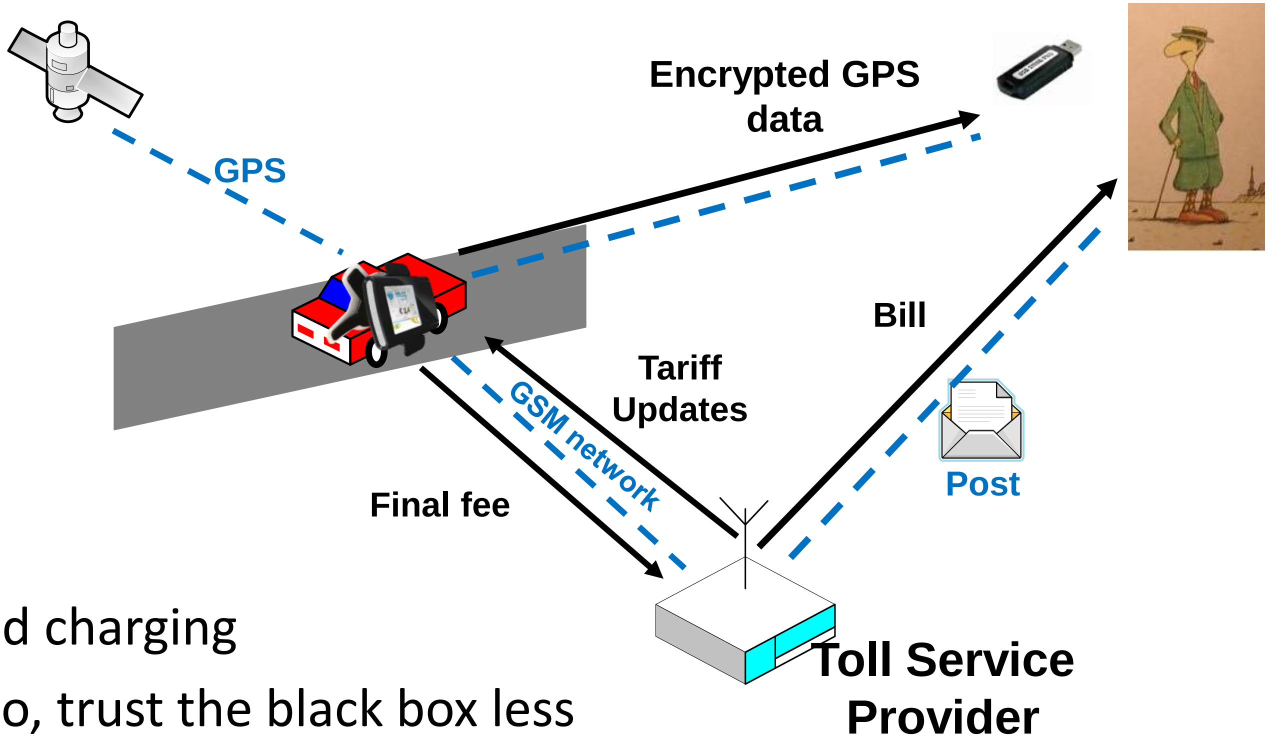
The payment database

Certified locations
for dispute resolution
given to user

PAYD & the devil in the detail

- Pragmatic design:
  - Against mass surveillance anyone with access to the car can violate privacy
  - Attacks against PriPAYD possible in PAYD (GPS, ...)
  - Disputes: trail available for user or default policy
  - Insurance holder gets to see where the car goes
    - Lesson for DPAs: extremely good is better than nothing
- Legal analysis of DP legislation
- Key objections: infeasible & expensive
  - Computations of GPS -> map -> segment -> price
  - 1 master student, 3 months, prototype ready, cost 10s of euros

PrETP for toll charging



- Hot: Tolls & road charging
- Use more crypto, trust the black box less

PrETP compared with PriPAYD

- Use of advanced cryptography to prove:
  - OBU was active, used correct prices, was at correct location, did all calculations correctly

- Enforcement mechanism?
  - Spot checks on road or using cameras
  - Present proof car was somewhere and expect opening of the charge for this location.

Holistic analysis

- From a theoretical point of view
  - The cryptography in the system ensures both privacy and law enforcement
- From a legal point of view
  - No personal data involved
  - Data minimization by design
- From a practical point of view
  - Prototype
  - Performance analysis
    - Computation
    - Communication

A warning against escrow

- PriPAYD & PrETP: illustrate how integrity, cheating detection, abuse resistance and privacy can be integrated

- Fallacy: "Why not store all the data and allow law enforcement to get access to it when this is legitimate?"
  - Undermines user confidence
  - Makes systems much more complex & expensive to engineer?
  - Prone to abuse: access control is poor, LI interfaces abused, key management is impossible

- Morality: if we want mass surveillance infrastructures we should build mass surveillance infrastructures
  - But, we must not turn privacy technologies into mass surveillance infrastructures

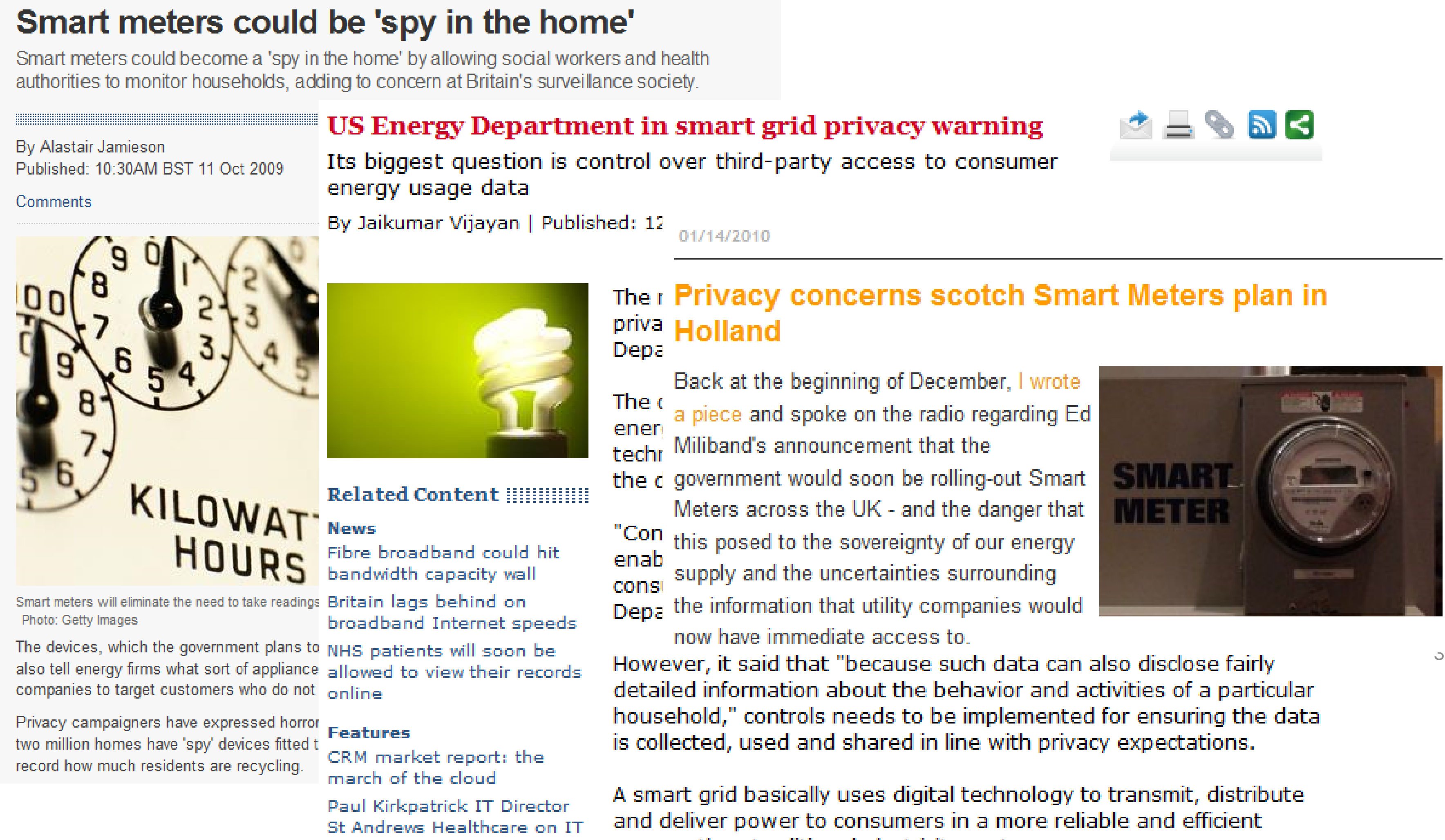
Privacy Technologies for the Smart Grid

George Danezis, University College London,

Alfredo Rial (KU Leuven / IBM), Klaus Kursawe (Nijmegen / ENCS), Markulf Kohlweiss (MSR), and Santiago Zanella-Beguelin (MSR), Andres Molina-Markham (UMass)

Jawurek, Marek, Florian Kerschbaum, and George Danezis. "Privacy technologies for smart grids-a survey of options."(2012).

First came the privacy controversy…



Smart meters could be 'spy in the home'

Smart meters could become a 'spy in the home' by allowing social workers and health authorities to monitor households, adding to concern at Britain's surveillance society.

By Alastair Jamieson
Published: 10:30AM BST 11 Oct 2009

Comments

Smart meters will eliminate the need to take readings
Photo: Getty Images

The devices, which the government plans to
also tell energy firms what sort of appliance
companies to target customers who do not

Privacy campaigners have expressed horror
two million homes have 'spy' devices fitted t
record how much residents are recycling.

US Energy Department in smart grid privacy warning

Its biggest question is control over third-party access to consumer energy usage data

By Jaikumar Vijayan | Published: 12

The r
priva
Depa

The c
ener
techr
the c

"Con
enab
cons
Depa

However, it said that "because such data can also disclose fairly detailed information about the behavior and activities of a particular household," controls needs to be implemented for ensuring the data is collected, used and shared in line with privacy expectations.

A smart grid basically uses digital technology to transmit, distribute and deliver power to consumers in a more reliable and efficient

Related Content

News

Fibre broadband could hit bandwidth capacity wall

Britain lags behind on broadband Internet speeds

NHS patients will soon be allowed to view their records online

Features

CRM market report: the march of the cloud

Paul Kirkpatrick IT Director St Andrews Healthcare on IT

01/14/2010

Privacy concerns scotch Smart Meters plan in Holland

Back at the beginning of December, I wrote a piece and spoke on the radio regarding Ed Miliband's announcement that the government would soon be rolling-out Smart Meters across the UK - and the danger that this posed to the sovereignty of our energy supply and the uncertainties surrounding the information that utility companies would now have immediate access to.

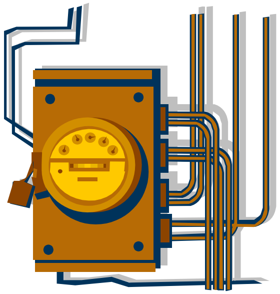
Smart meters for electricity

Certified & Tamper resistant

Utilities & Others

Measures power consumption
Over every 15-30 minutes (KW/h)

Registers for input /
output (micro generation)
and multiple channels



- Real time aggregates
- Time-of-use billing
- Forecasting
- Fraud detection

Remote disconnection

Pre-payment

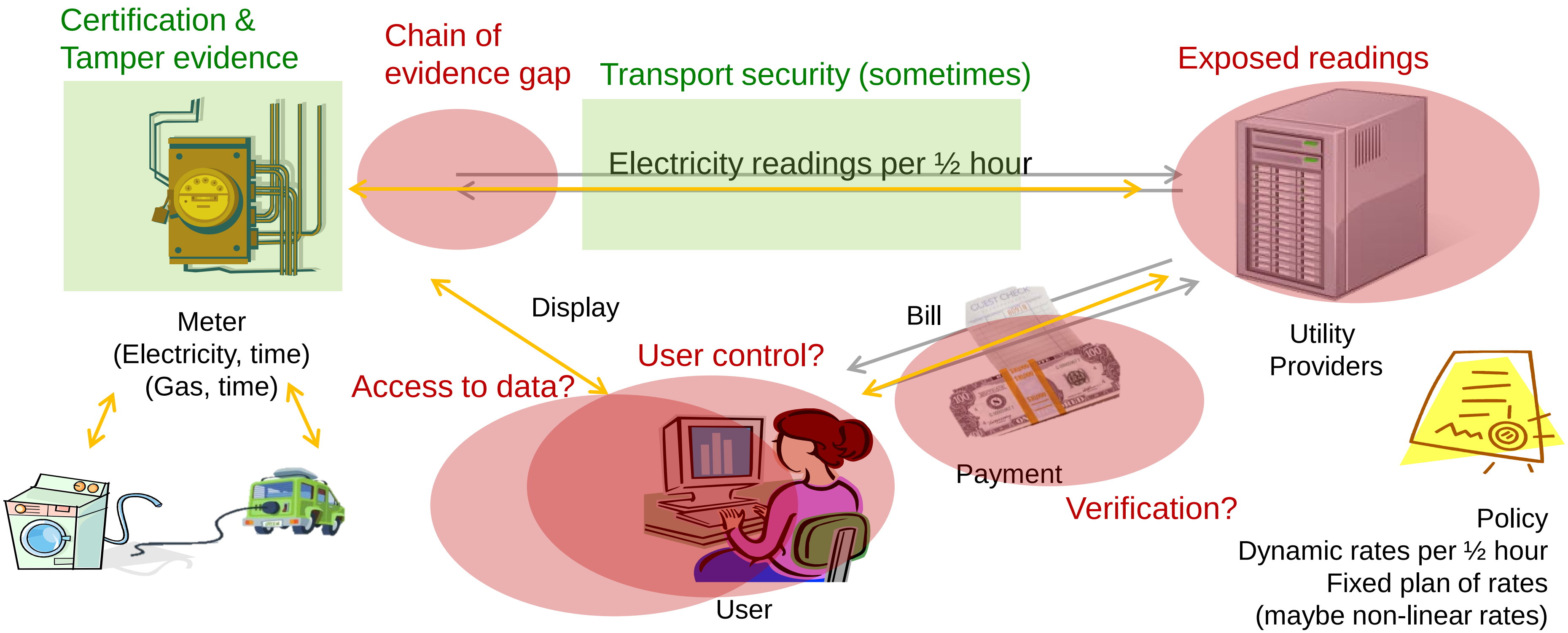
Demand-Response

Stores readings for
up to 13 months.

Wide area network
Communications for
control and readings

Lesson: Infrastructure security is in a multi-stakeholder an open ended use context.

Metering Security & privacy today



Certification & Tamper evidence

Chain of evidence gap

Transport security (sometimes)

Exposed readings

Electricity readings per ½ hour

Meter
(Electricity, time)
(Gas, time)

Display

Bill

Access to data?

User control?

Utility Providers

Payment

Verification?

User

Policy
Dynamic rates per ½ hour
Fixed plan of rates
(maybe non-linear rates)

Desirable properties for 21st century "legal metrology"

- Utility:
  - Enable computations on readings
  - Aggregate statistics in real-time
- Security:
  - End-to-end authenticity and integrity
  - Privacy / information self-determination
- Versatility & public policy objectives
  - Infrastructure
  - Platform
- Robust security engineering

Security properties

Authenticity & Integrity

- **End-to-end** property:
  - Establish the authenticity & integrity of a reading throughout the life time of the reading.
  - "Valid reading from specific certified metrology unit".
- **Universal** / public verifiability:
  - No need for secrets to verify readings -- all parties can verify them.
- Stronger: **integrity of computations**:
  - Interaction with privacy = not trivial.
  - **Software independence** = no chain of custody / can use untrusted hardware.

Privacy & self-determination

- Household readings are **personal data** (DP!)
  - Allow inferences on occupancy, lifestyle, health, religion, …
- Privacy as **confidentiality**:
  - Gold standard: only data subject has access to raw readings.
  - **Data minimization**: e.g. private aggregation.
  - But: others should still be able to **compute** on them.
- Privacy as informational **self-determination**:
  - Subject can **use readings further** with 3rd parties.
  - **Audit computations** performed on personal data.
  - Use **any device / OS**.

Other goals

Public policy

- Meters as part of **platform & Infrastructure**
  - Need for versatility, extensibility, choice.
  - Lifetime: open to future technologies.
- Support **competition**:
  - No lock-in for any party.
  - High-quality readings for all.
  - Ability to use any user device.
- Support **secondary uses**.
  - Aggregation: with privacy.
- **Need for standardization!**

Robust security engineering

- Minimal Trusted Computing Base
  - Minimal trusted hardware
  - Ideally: just the **certified metrology unit**.
  - Amenable to **formal verification**.
- Trusted third parties
  - Ideally: **no single TTP**
  - 4C: Cost, Collusion, Compulsion, Compromise.

How to combine privacy and integrity?

- Naïve architecture:
  - Encrypt all readings. Store them encrypted. Authentication and authorization to distribute keys to 3rd parties that *decrypt* to compute and aggregate.

- Counter-intuitive architecture:
  - Encrypt all readings. Store them encrypted. Aggregation without decrypting readings, and Computations on client device (with no cheating!)
  - Readings are never decrypted outside user device.
  - Aggregation can occur in real-time without leaking raw readings.

- Or any mixture of both …

Privacy technology ingredients

- Special signature scheme & architecture
  - Allows for end-to-end integrity & authenticity + privacy friendly computations.

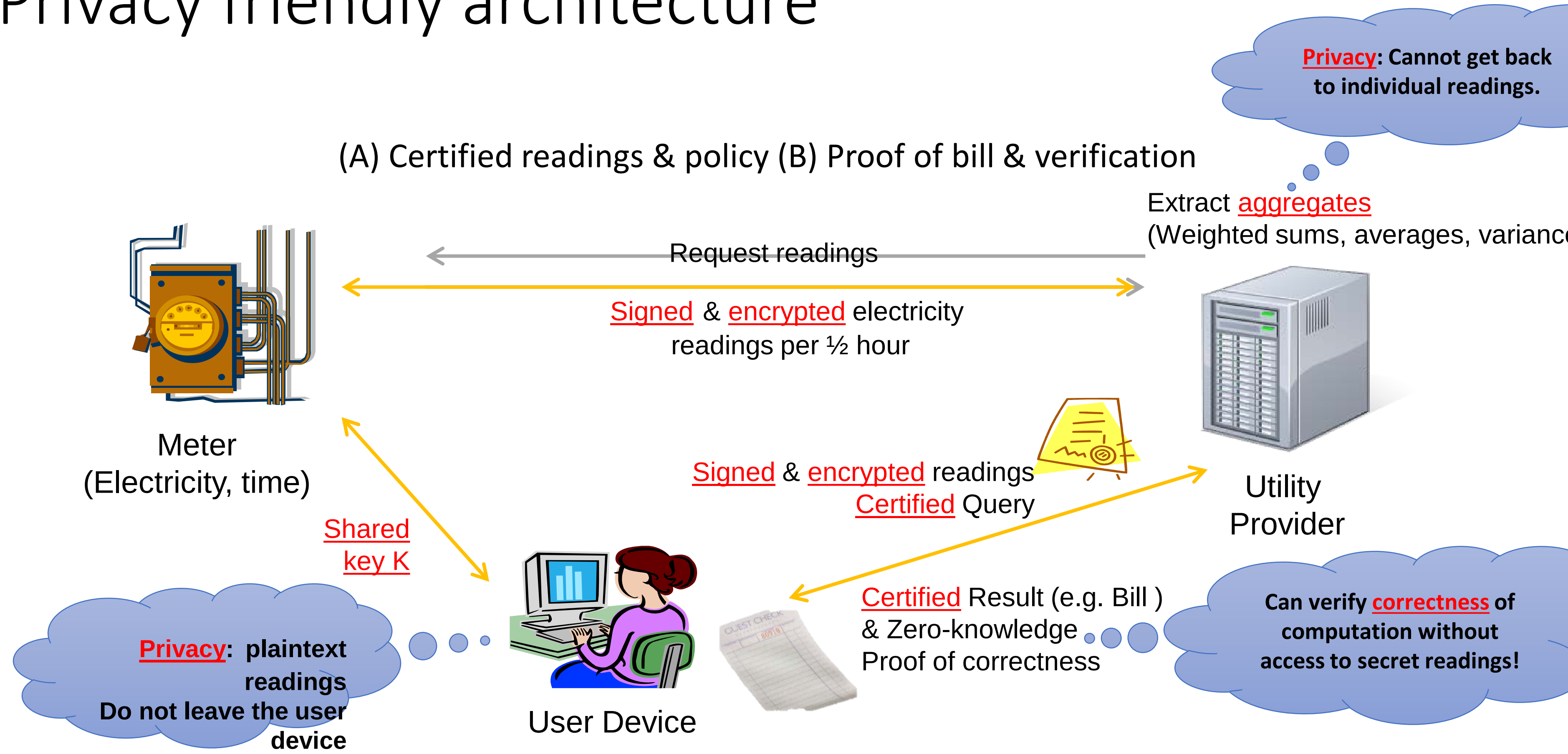
Hint: Sign Pedersen commitments of readings.

- Special encryption scheme
  - Allow for aggregation from ciphertexts to get statistics.

Hint: Blind readings with shares from other meters or authorities.

- Standard zero-knowledge techniques + language support
  - Perform computations on user machines while preserving privacy and integrity.

Privacy friendly architecture



Privacy: Cannot get back to individual readings.

(A) Certified readings & policy (B) Proof of bill & verification

Extract aggregates
(Weighted sums, averages, varianc

Request readings

Signed & encrypted electricity readings per ½ hour

Meter
(Electricity, time)

Signed & encrypted readings
Certified Query

Utility
Provider

Shared key K

Certified Result (e.g. Bill)
& Zero-knowledge
Proof of correctness

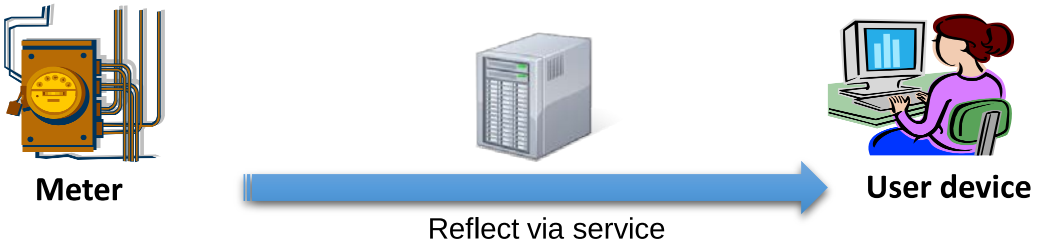
Can verify correctness of computation without access to secret readings!

Privacy: plaintext readings
Do not leave the user device

User Device

Alfredo Rial and George Danezis. **Privacy-preserving smart metering**. In Yan Chen and Jaideep Vaidya, editors, WPES, pages 49-60. ACM, 2011

Details on generating and consuming readings



- Each meter has a secret key K_m
- For each reading (t, r) …
- Generate a Pedersen commitment to r
  - $C_r = g^r h^o$ for an $o = Hash_o(K_m ; t)$
- Hash the commitment into
  - $H = Hash(H ; C_r)$
- Throw the commitment away!
- Encrypt reading as
  - $t, c = t, (r + s) \bmod 2^{32}$
- Sign over encrypted readings and H
- Send times, enc. readings and signature to service.

Hiding & Binding

What is “s”?
Could be a function of the key

- Get secret K_m from meter
- Recover readings as
  - $r = (c – s) \bmod 2^{32}$
- Re-compute $o= Hash_o (K_m ; t)$ and $C_r = g^r h^o$
- Send all commitments C_r to service
- Use commitments C_r to prove anything about the committed values using Zero-Knowledge proofs.

What are those?

Service can check signature to validate commitments to readings.

Efficiency considerations

Meter networking:

- Plaintexts size = Ciphertext size = 32 bits.
- Signature is augmented, but size is the same.
- No need to modify DLMS protocols & stress the 9600 bps modems.
- Larger commitments not sent by the meters (saves about 160 bits / reading).

- *From user device to service there is plenty of bandwidth.*

Meter computations:

- Commitment: h^o can be pre-computed. Then g^r in fewer than 32 multiplications. Or 4 multiplications with lookup tables of 256 values.
- Hash function computations are very fast, and so is addition.
- Experiments: on real meters, ARM Cortex 7, and 8 bit microcontrollers.
- Code + memory size the key bottleneck (strip down SSL bn).

- *User device assumed to be more powerful and have more memory.*

Andres Molina-Markham, George Danezis, Kevin Fu, Prashant J. Shenoy, and David E. Irwin. **Designing privacy-preserving smart meters with low-cost microcontrollers.** In Financial Cryptography 2012, LNCS 7397, pages 239-253.

Two flavours of computations on user device

- Fast linear computations (Billing protocol):
  - Special case: policy is public, and selection of rate independent of reading.
  - Very fast: process 3 weeks of all UK data in 12 days on 1 CPU core.

With ZK proofs of correctness

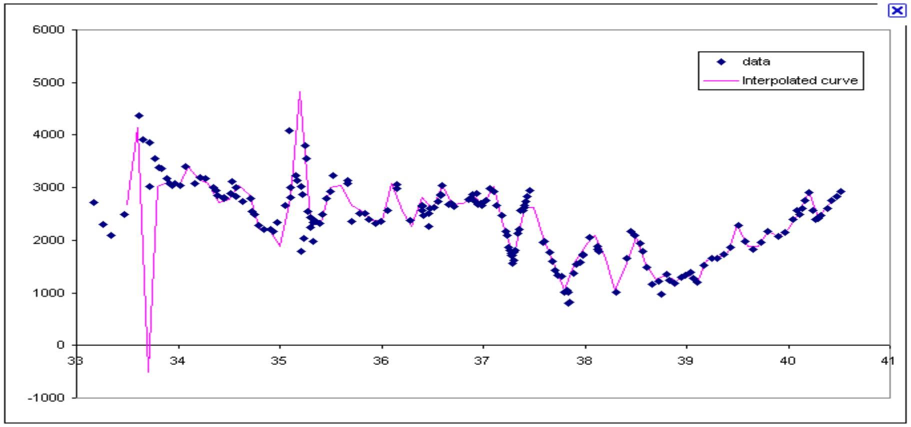
- Generic computations protocol:
  - Uses more expensive generic Zero-Knowledge proofs.
  - ZKP: allows one to prove any relation on committed values holds!
  - In theory supports any computation (some faster than others)

General computations?

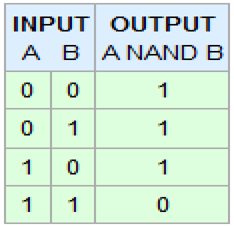
- Fast protocol:
  - Linear algebra: $Result = \Sigma_i\, x_i \cdot r_i$
- General zero-knowledge proofs:
  - Multiplication $Result = x_i \cdot r_i$
  - Lookup: $Result = Table[\, r_i \,]$
  - Range: $Result = Table[\, min < r_i < max]$
  - Polynomial: $Result = a\, r_i{}^3 + b\, r_i$

Any function!

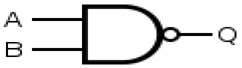
- Ranges + polynomials
  = splines
  = any function



- “*” or Table[]
  = NAND gate
  = any circuit

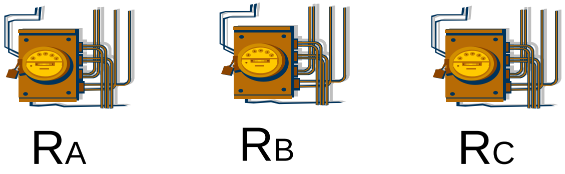


INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



Any circuit = Slow!

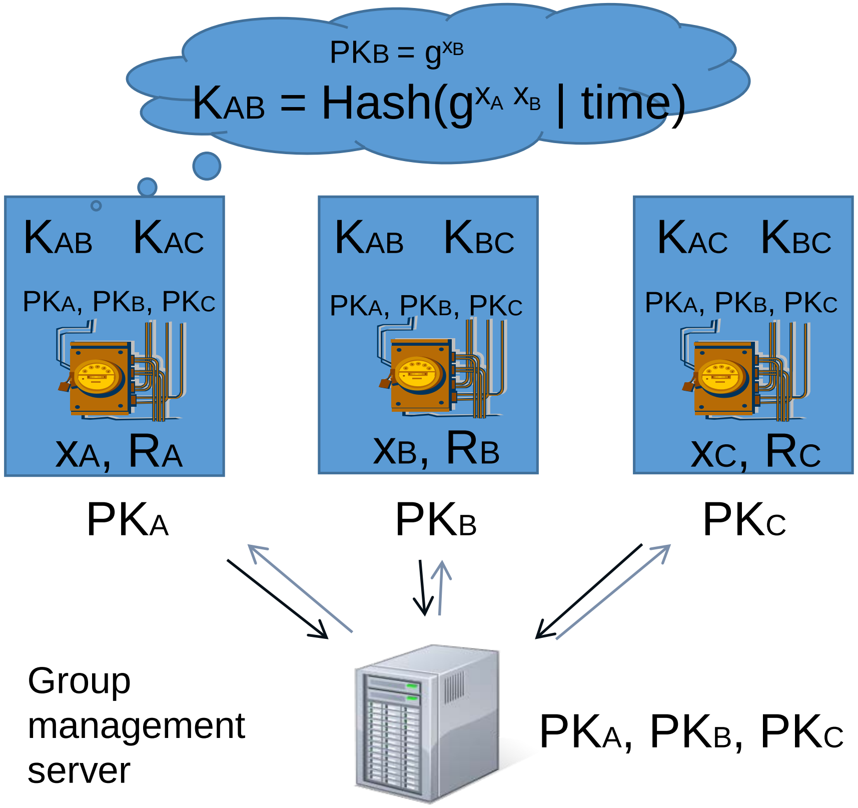
Privacy friendly aggregation



- Aim: compute sum without revealing readings.
- 2 Phases:
  - Distribute keys
  - Compute readings

Klaus Kursawe, George Danezis, and Markulf Kohlweiss. **Privacy-friendly aggregation for the smart-grid**. In Simone Fischer-H□ubner and Nicholas Hopper, editors, PETS, volume 6794 of Lecture Notes in Computer Science, pages 175-191. Springer, 2011

Privacy friendly aggregation



$PK_B = g^{x_B}$

$K_{AB} = \text{Hash}(g^{x_A x_B} \mid \text{time})$

K_{AB} K_{AC}

PK_A, PK_B, PK_C

x_A, R_A

K_{AB} K_{BC}

PK_A, PK_B, PK_C

x_B, R_B

K_{AC} K_{BC}

PK_A, PK_B, PK_C

x_C, R_C

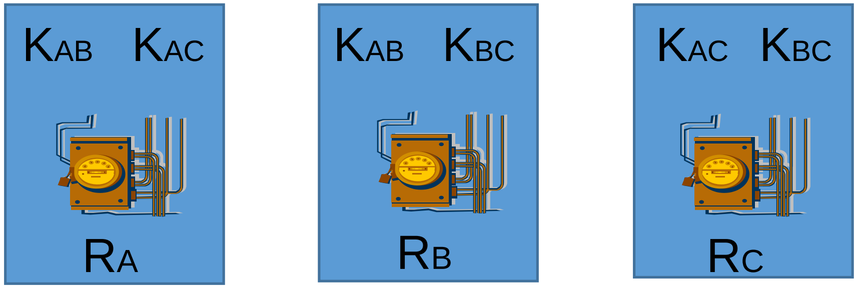
PK_A PK_B PK_C

Group management server

PK_A, PK_B, PK_C

- Aim: compute sum without revealing readings.
- 2 Phases:
  - **Distribute keys**
  - Compute readings

Privacy friendly aggregation



- Aim: compute sum without revealing readings.
- 2 Phases:
  - Distribute keys
  - **Compute readings**

$C_A = R_A + K_{AB} + K_{AC}$

Group management server

$C_B = R_B - K_{AB} + K_{BC}$

$C_C = R_C - K_{AC} - K_{BC}$

This is “s”!

t, c = t, (r + s) mod 2^{32}

$\text{Sum} = C_A + C_B + C_C = R_A + R_B + R_C$

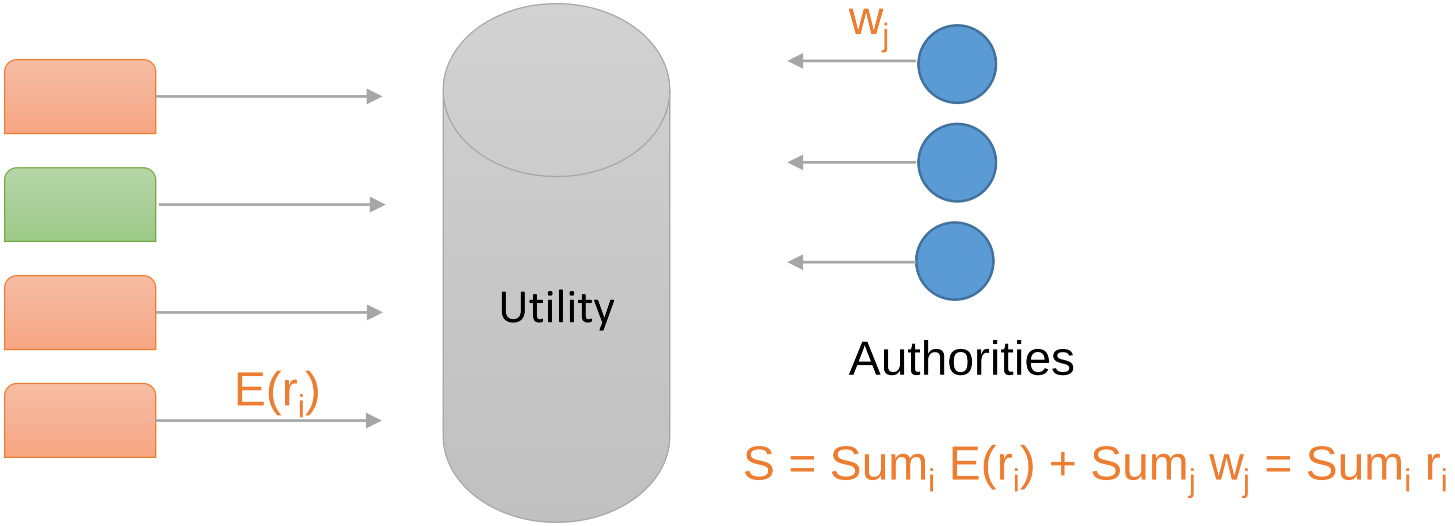
Better aggregation protocols?

- Problems with simple aggregation protocol:
  - Robustness: what if some meters fail?
  - Flexibility: Can I define different groups?
  - Generality: Weighted sums?
  - Privacy: what about repeated queries? What about side information?

- Yes! We can do all this, but:
  - We need a set of trusted non-colluding authorities
  - They do not learn any secret.
  - Weighted sums, auditing, differential privacy, linear time of use bills!

Gilles Barthe, George Danezis, Benjamin Gregoire, Cesar Kunz, Santiago Zanella-Beguelin.
Verified Computational Differential Privacy with Applications to Smart Metering. Accepted in IEEE Computer Security Foundations, 2013.

Robust Private Aggregation



$S = \text{Sum}_i\ E(r_i) + \text{Sum}_j\ w_j = \text{Sum}_i\ r_i$

- Problem: A missing reading makes the aggregate unavailable.
- Solution: Rely on a number of authorities to decrypt instead.
  - Any subset of meters can be aggregated.
  - K-out-of-N authorities are needed.
  - Can compute any weighted sum of meter reading: aggregates & bills.
- TCB? No single place for confidentiality.

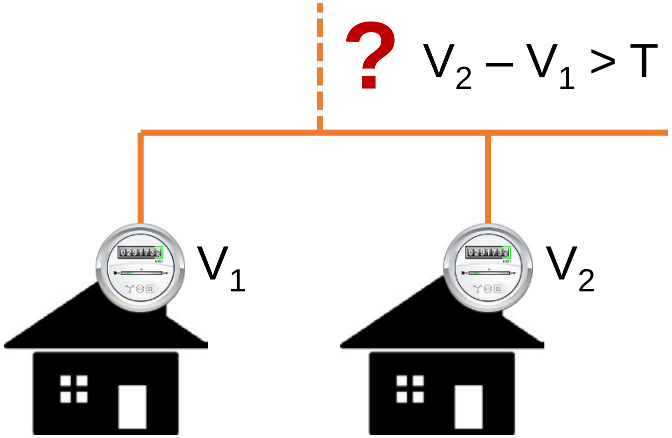
Barthe, Gilles, George Danezis, Benjamin Grégoire, César Kunz, and Santiago Zanella-Béguelin. "**Verified computational differential privacy with applications to smart metering**." In Computer Security Foundations Symposium (CSF), 2013.

The need for differential privacy

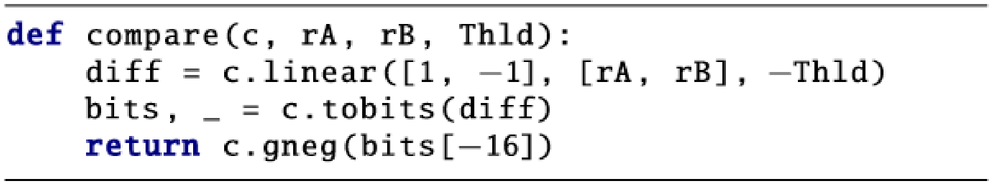
- Simple aggregation protocol:
  - fixed groups = fixed computations / time.
- Flexible aggregation protocol:
  - Arbitrary linear sums of secrets.
  - Attack: n-queries can extract n exact secrets.
- Solution:
  - Distributed differentially private mechanism.
  - Each authority adds some noise.
  - 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



Line theft detection use-case



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

- 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.

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

Further work

- More paranoid?
  “What about the information leaking from billing?”
  - We can inject positive noise to bills, to ensure differential privacy for a certain number of billing periods.
  - This is expensive in the long term.
  - No worries! we can also keep a secret tally of the additional expense, and reclaim that money in the future!
- More sophisticated computations?
  “But I want to run a classifier on the readings, to profile your household!”
  - Lookup tables allow for efficient ML classifiers
  - We can prove the classification result from Random Forests & HMMs

George Danezis, Markulf Kohlweiss, and Alfredo Rial. **Differentially private billing with rebates.** In Information Hiding, LNCS 6958, pages 148-162. Springer, 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

In conclusion

- **Smart Metering could be implemented to accommodate strict privacy policies and provide very high integrity.**
  - Flexible computations with privacy and integrity on the client side.
  - Real time, flexible and robust aggregation.
- Lack of privacy by design can jeopardise industrial deployments; so can poor security engineering.
  - Netherlands: failure through lack of privacy thinking.
  - Germany: failure due to mandating inappropriate solutions.
- The gap between cryptographic theory and security engineering is vast, but can be bridged.
  - Fitting expensive primitives within constrained devices, and environments.
  - ZQL, and other tools to allow non-expert security engineers to be productive.
- Solutions still require some paradigm shifts:
  - Trustworthy computations in the client domain for privacy – removes some control.
  - Private aggregation as a key service for big data analytics – a business opportunity.