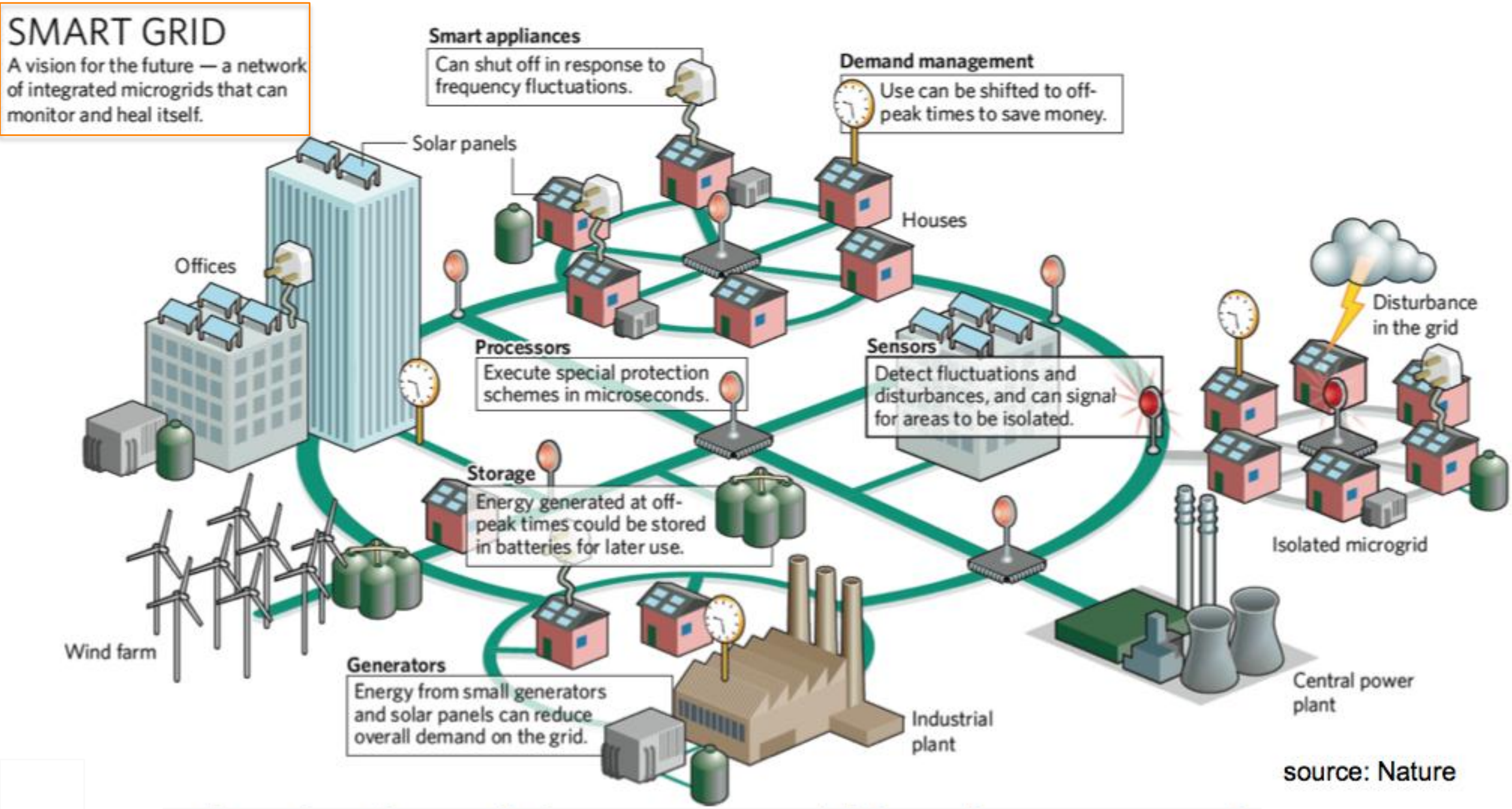
A DATA PSEUDONYMIZATION PROTOCOL FOR SMART GRIDS

Giulia Mauri

22-05-2013



source: Nature

Introduction (II)

- Detailed energy consumption measurements allow:
  - ✔ Timely management of energy distribution,
  - ✔ Efficient grid monitoring,
  - ✔ Energy forecasting and provisioning.
  - ✖ Inference of customers' personal habits,
  - ✖ Identifying and tracking customers,
  - ✖ Exposing customer behaviour for commercial benefits.

Introduction (III)

- Why anonymizing metering data?

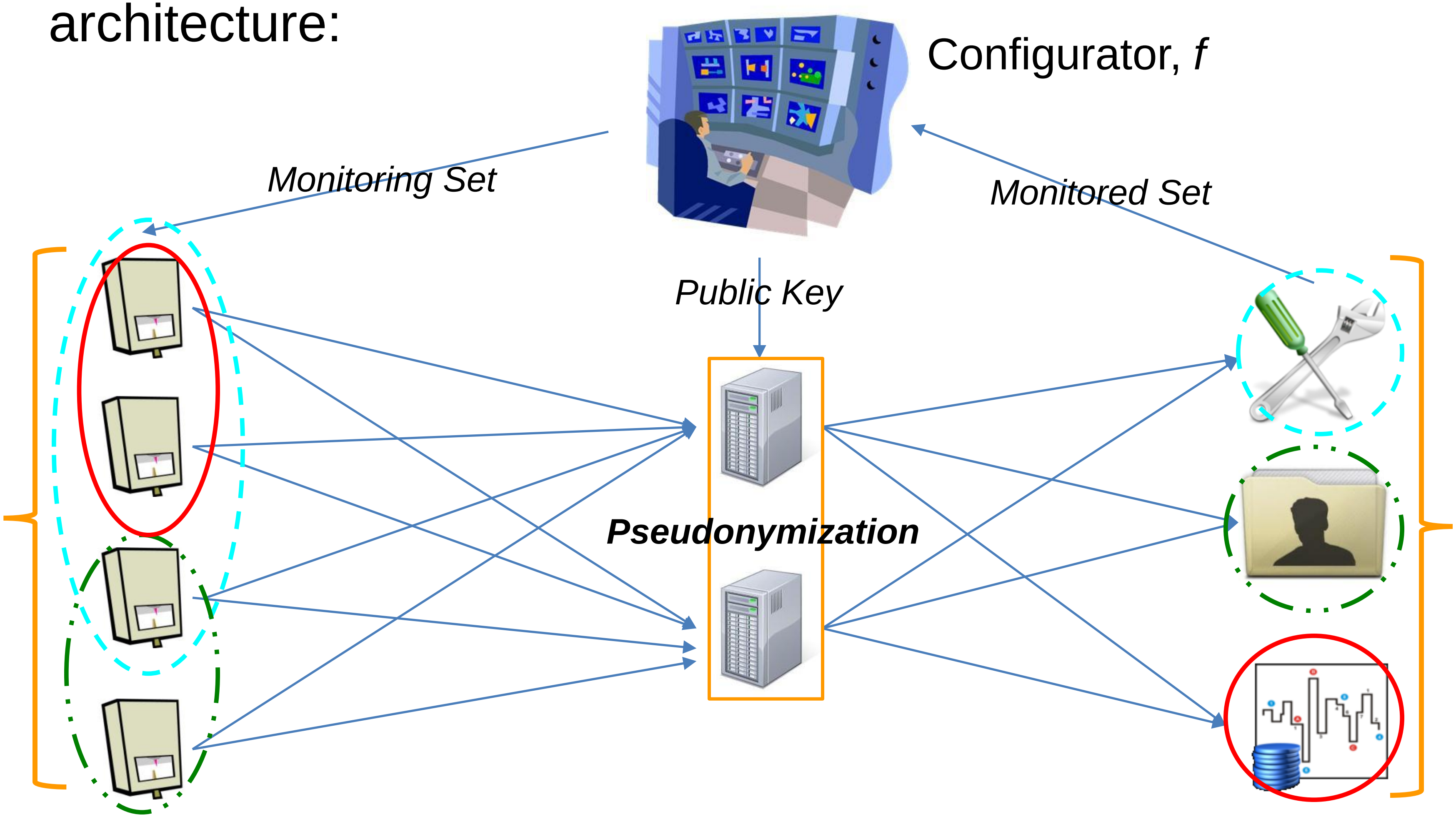
  According to NIST, “Smart Grid data should be anonymized wherever possible to limit the potential for computer matching of records.”

- Different approaches have been proposed for data anonymization, including:
  - ✓ Generalization
  - ✓ Perturbation
  - ✓ Pseudonimization
  - ✓ Aggregation

Introduction (IV)

- Some open problems need to be solved:
  - ✓ Multiple data Consumers,
  - ✓ Low computational load on Meters,
  - ✓ Frequent re-pseudonymization,
  - ✓ Identity recovery, if necessary.

- Three different sets of nodes are comprised in proposed architecture:



1. Producers, *p*
2. PPNs, *n*
3. Consumers, *c*

- The pseudonymization protocol consists of a tuple of algorithms:

✓ $\texttt{Setup}(1^l) \rightarrow (k_d, params)$

✓ $\texttt{pSend}(param, i, p, x_i^p) \rightarrow \left(e_i^p(1), \ldots, e_i^p(n), \ldots, e_i^p(N), ID_p, r_i^p\right)$

✓ $\texttt{PPNSend}(param, i, n, ID_p, r_i^p, e_i^p(n)) \rightarrow \left(PD_c^p, e_i^p(n)\right)$

✓ $\texttt{cReceive}(param, i, c, PD_c^p, e_i^p(1), \ldots, e_i^p(N)) \rightarrow \left(PD_c^p, x_i^p\right)$

- The encryption algorithm used in `pSend` is the Shamir Secret Sharing Scheme, that we assume to be *unconditionally secure.*

- Full Pseudonymization:

$$\Pr(full - p = 1) \leq \frac{1}{2} + negl(l)$$

- Full Pseudonymization with Perfect Forward Anonymity:

$$\Pr(full - p - pfa = 1) \leq \frac{1}{2} + negl(l)$$

- Unconditionally Indistinguishable Encryption:

$$\Pr(blind = 1) = \frac{1}{2}$$

Security Properties
The full-p experiment

- The experiment for an algorithm *A* and a parameter *l*, assumes an adversary Consumer *c**, and focuses on two Producers *ID1,ID2*.

✓ The Setup outputs the system parameters.

✓ The first and second Producers execute pSend and output the messages: $msg_1^1 \dots msg_N^1, msg_1^2 \dots msg_N^2$.

✓ Each PPN receives the two messages, and calls the PPNSend. Then each PPN sends two messages $pmsg_n^p$ with $p \in \{1,2\}$ to the Consumers.

✓ Finally each Consumer runs cReceive and obtains the measurement with the pseudonym.

✓ The malicious Consumer *c** executes *A* and outputs $p' \in \{1,2\}$.

✓ The output of the experiment is 1 if $p' = p$, and 0 otherwise.

Pseudonymization Function

- We use a keyed one-way function with trapdoor $\mathbf{E_{Ke}(m,r) = y}$ with the following properties:

✓ The Configurator generates public/private key pair and keeps the private key and distributes the public key.

✓ Our implementation of $\mathbf{E_{Ke}}$ builds upon *RSA with OAEP*.

✓ Each PPN calculates the pseudonym as:

$$PD_c^p = E_{k_e}[ID_p||c||\left\lceil \frac{i}{\alpha} \right\rceil \alpha, w_p^c]$$

- **ID_p**: Producer's Identity
- **c**: Consumer's Identity
- **i**: round identifier
- **α**: length of the validity time span of pseudonym
- **w^c_p**: security nonce

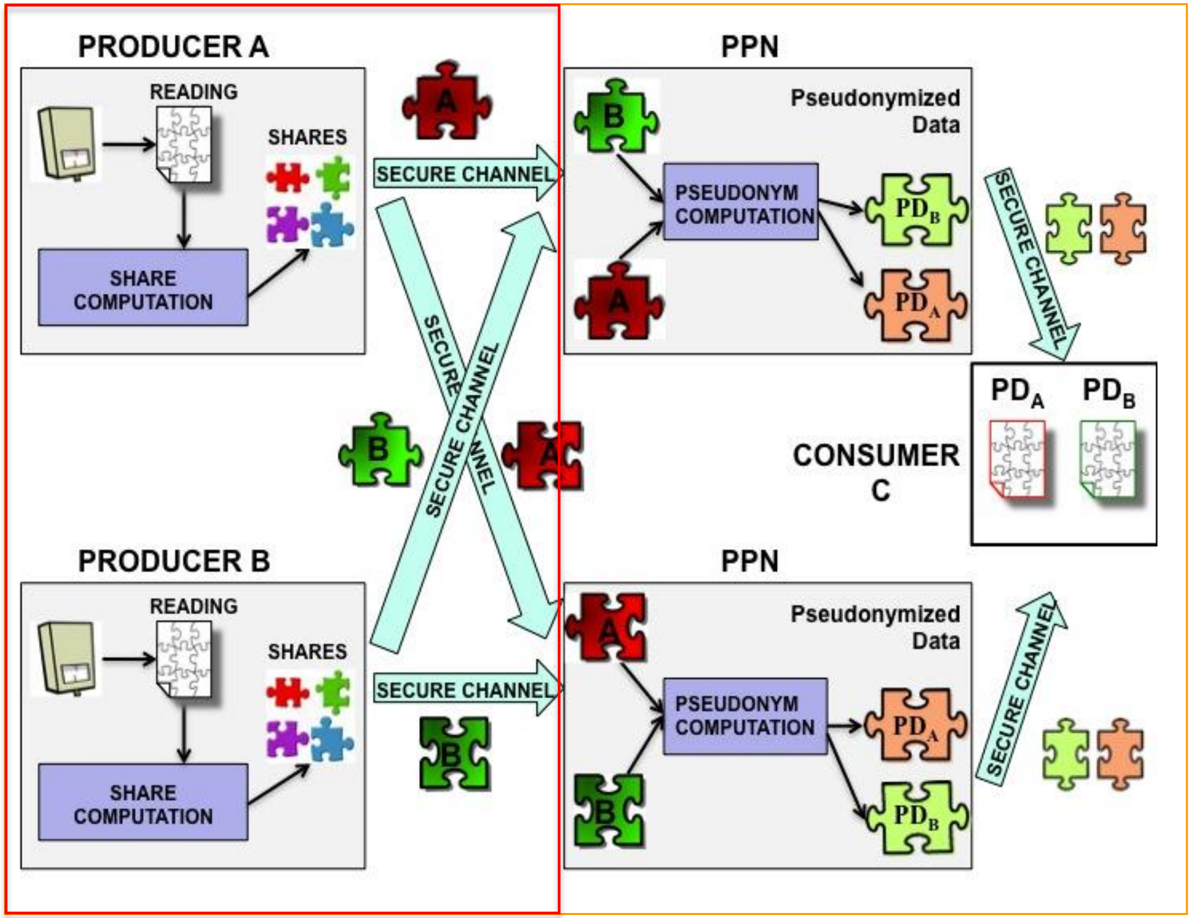
Other security properties

- There exists a p.t. algorithm that, given the private key, can **recover the identity** of Producer from its pseudonym.
  - ✓ This property is consequence of Configurator having the private key.
- Before sending its data, the **Producer is aware** of the set of Consumer monitoring its data.
  - ✓ This happens thanks to the message *SpecifyMonitoringSet*.
- Given a pair of distinct Producers' identities (p,p') and the same Consumer c, or a pair of distinct Consumers (c,c') and the same Producer p, the **output** of the function E_{k_e} is **always different**.
  - ✓ This property is consequence of using the ciphering function that relies on RSA with OAEP.

Communication Protocol (I)
Shamir Secret Sharing Scheme

At every round, the following procedure is repeated:

1. Producer *p* divides its measurements in *t* shares and gives one share to each of the *t* PPNs.

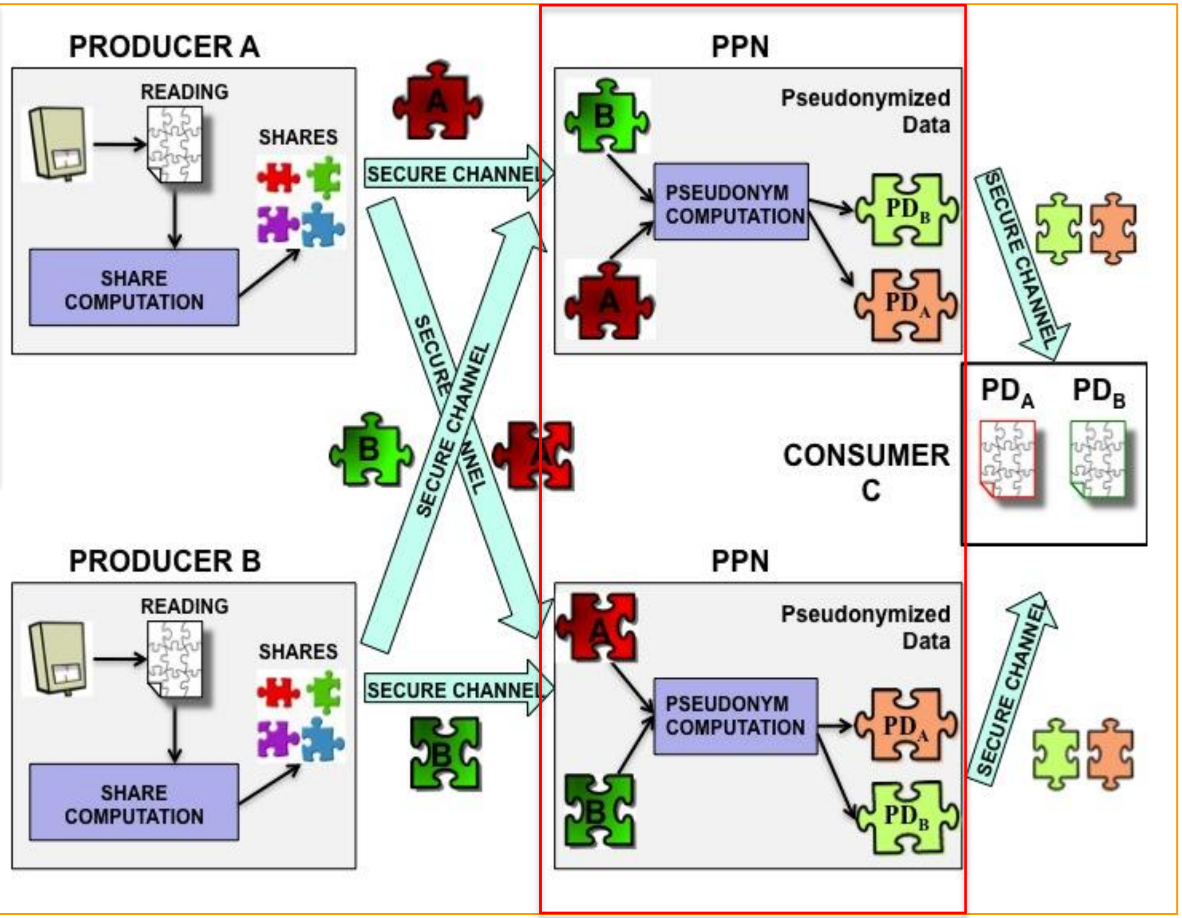


Communication Protocol (II)
Shamir Secret Sharing Scheme

At every round, the following procedure is repeated:

1. Producer *p* divides its measurements in *t* shares and gives one share to each of the *t* PPNs.

2. PPN *n* receives a share from Producer *p* destined to Consumer *c*, computes the pseudonym basing on *p* and *c* and sends it to *c* together with the share.



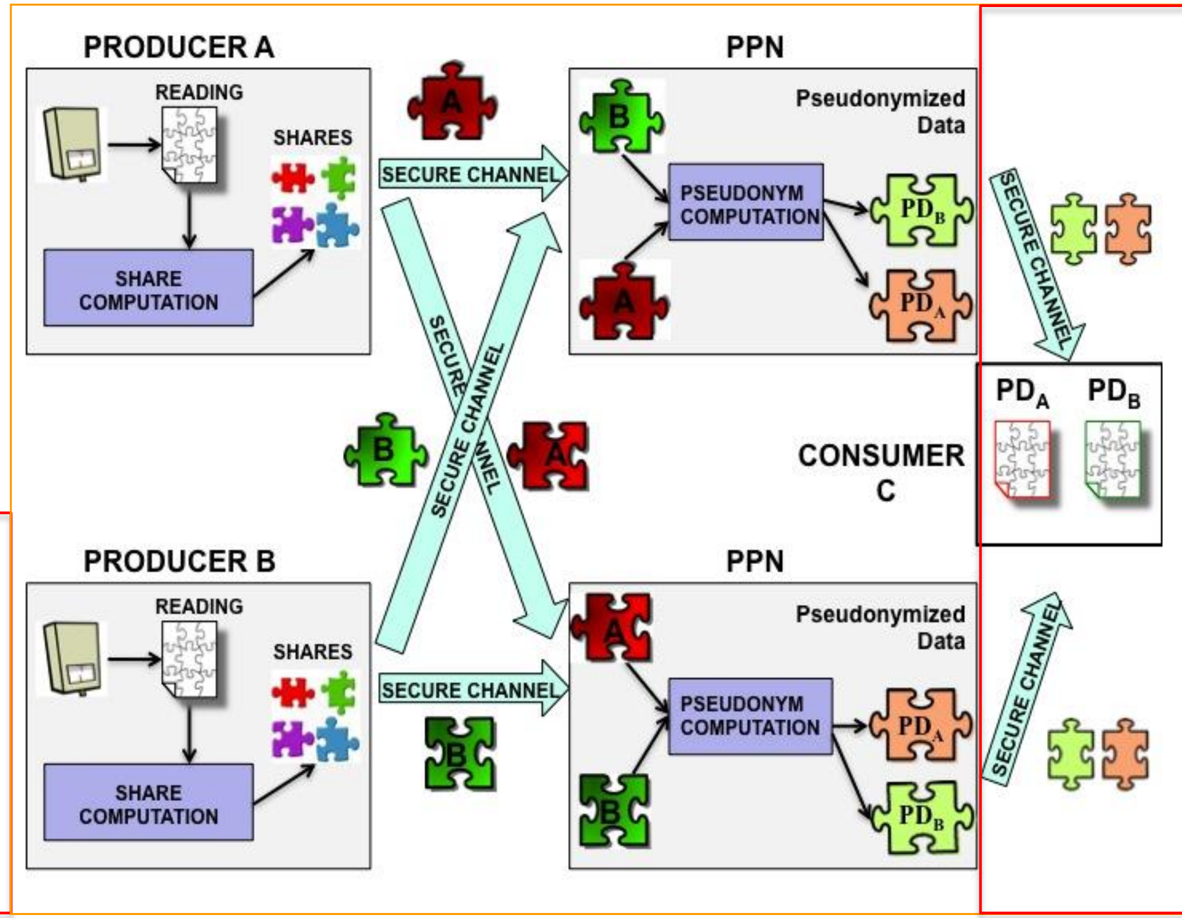
Communication Protocol (III)
Shamir Secret Sharing Scheme

At every round, the following procedure is repeated:

1. Producer *p* divides its measurements in *t* shares and gives one share to each of the *t* PPNs.

2. PPN *n* receives a share from Producer *p* destined to Consumer *c*, computes the pseudonym basing on *p* and *c* and sends it to *c* together with the share.

3. Consumer *c* combines the shares associated to the same pseudonym and recovers the measurements.



Comparison and Results

- We compare the performance of our proposed protocol with:

✓ Mixing Scheme: it moves the computational load on the Producer that computes its pseudonym and creates the mixing packet. The Producers encrypt with RSA and the PPNs only forward messages.

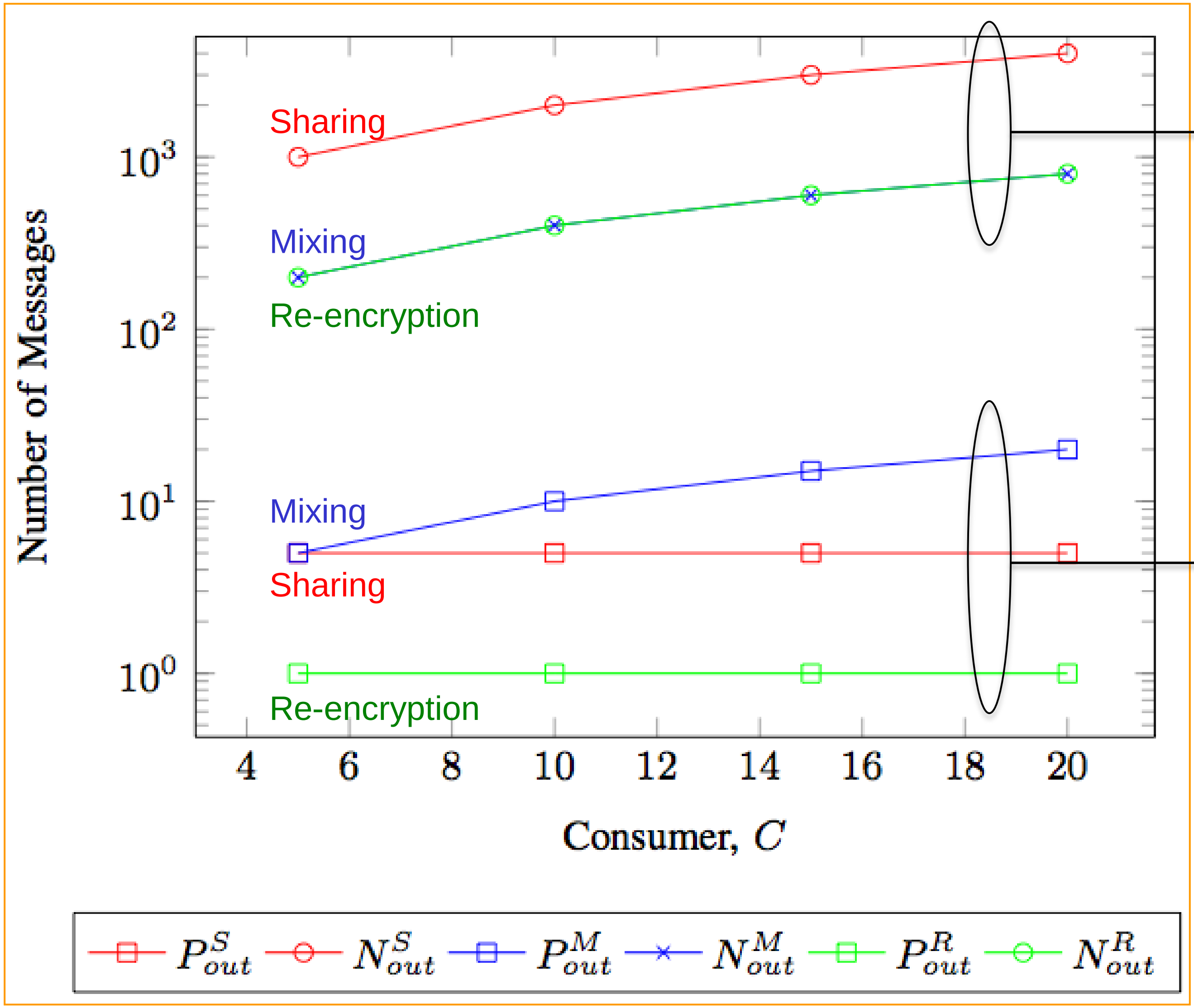
✓ Proxy Re-Encryption Scheme: it guarantees that a collusion of all PPN can't obtain the relation between Producer's measurement and identity. The Producers and PPNs encrypt with Paring based algorithm.

- We evaluate the number of sent messages and the computational cost in the three scheme.

Results (I)
Number of Sent Messages

Asymptotic Values	Mixing Scheme	Shamir Secret Sharing	Proxy Re-Encryption
Messages sent by each Producer	|C|	|N|	1
Messages sent by each PPN	|P|*|C|/|N|	|P|*|C|	|P|*|C|/|N|

Results (I)
Number of Sent Messages



Number of messages sent by each PPN

Number of messages sent by each Producer

|P|=200, |N|=5

Results (II)
Computational Costs

Asymptotic Values	Mixing Scheme	Shamir Secret Sharing	Proxy Re-Encryption
Producer	|C|* Cost(RSA_{enc})	Cost($Share_{gen}$)	Cost(Pairing)
PPN	|P|*|C|/|N|* Cost(Forward)	|P|*|C|* Cost(Forward)	|P|*|C|/|N|* Cost(Pairing)
Consumer	|P|*Cost(RSA_{dec})	|P|*Cost($Share_{join}$)	2*|P|* Cost(Pairing)

Results (II)
Computational Costs

C=10	Mixing	Sharing	Re-Encryption
Producer	5,12 ms	0,10 ms	21,43 ms
PPN	-	-	> 5 min
Consumer	4,86 s	2,11 s	> 5 min
C=50	**Mixing**	**Sharing**	**Re-Encryption**
Producer	25,60 ms	0,10 ms	21,43 ms
PPN	-	-	>> 5 min
Consumer	4,86 s	2,11 s	> 5 min
C=100	**Mixing**	**Sharing**	**Re-Encryption**
Producer	51,20 ms	0,10 ms	21,43 ms
PPN	-	-	>> 5 min
Consumer	4,86 s	2,11 s	> 5 min

$|P|=1000, |N|=5$

Processor: 2.7 GHz Intel Core i7, Memory: 4GB 1333 MHz DDR3

Conclusion

- ✓ We propose a pseudonymization protocol for smart metering measurements,
- ✓ The protocol allows collecting metering data without revealing the association between users' identities and their pseudonyms,
- ✓ We described a possible implementation of the proposed algorithm,
- ✓ This work evaluates the security guarantees and the performance the algorithm achieves.
- ✓ We compare three different solutions in terms of number of sent messages and computational costs.

➔Results show that the most suitable protocol is the one based on pseudonymization with the Shamir secret sharing scheme.


THANK YOU
© Giulia Mauri

Appendix: Related Work (I)

- Solutions proposed for protecting user data in AMI (Advanced Metering Infrastructure):
  - ✓ Zero-knowledge cryptographic protocols [1],
  - ✓ Data aggregation [2],
  - ✓ Escrow services [3].
- Our solution has been proposed in:
  - ✓ C. Rottondi, G. Mauri and G. Verticale, " A data pseudonymization protocol for smart grids", in *IEEE OnLine Conference on Green Communication*, 2012.

[1] A. Rial and G. Danezis, "Privacy-preserving smart metering," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, ser. WPES 2011.

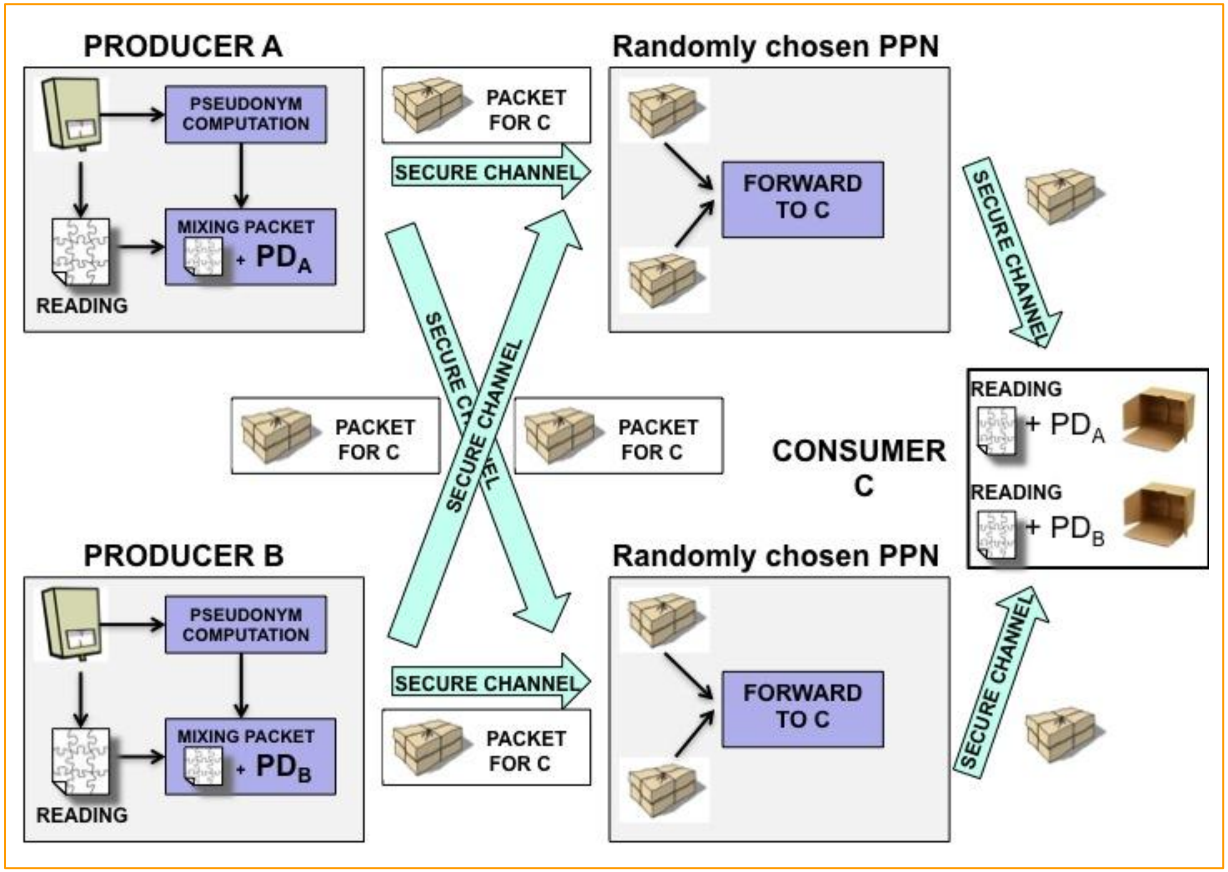
[2] C. Rottondi, G. Verticale, and A. Capone, "A security framework for smart metering with multiple data consumers," in *First IEEE INFOCOM CCSES Workshop on Green Networking and Smart Grids*, 2012.

[3] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010

Mixing Scheme

At every round, the following procedure is repeated:

1. Producer *p* generates the measurements and computes its pseudonym. It creates the mixing packet, composed by measurement and pseudonym, that is sent to a randomly chosen PPN *n*.



2. PPN *n* forwards the packet to the Consumer *c*, to whom the message is destined.

3. Consumer *c* recovers the individual data by decrypting the packets.

At every round, the following procedure is repeated:

1. Producer *p* encrypt its measurements based on the identity of PPN, to whom the message is destined.

2. PPN *n* computes the pseudonym basing on *p* and *c*, re-encrypt the packet and sends it to *c*

3. Consumer *c* recovers the individual data by double decrypting the packets.