Membership Inference Attacks on Aggregated Time Series with Linear Programming

Antonin Voyez[1, 3], Tristan Allard[1], Gildas Avoine[1, 2], Pierre Cauchois[3], Élisa Fromont[1, 4, 5], Matthieu Simonin[4]

[1]Univ Rennes, CNRS, IRISA, France [2]INSA Rennes, CNRS, IRISA, France [3]ENEDIS, France [4]Inria, IRISA, France [5]IUF (Institut Universitaire de France), France

Context

Publication context

- Smart meters are devices used to collect fine grained electricity consumption data.
  - France: 34M meters, USA: 100M meters, ...
- European legislations[1] strongly encourage the collection and publication of energy consumption time series.
  - Ecological transition (thermal renovation)

Publication methods

- State of the art protection methods (differential privacy) are not always used in real life.
  - Differential privacy is hard to apply to unbounded time series.
  - Ex: Threshold-based aggregation (sum / average of several meters per timestamp) in France.

[1] *Directive 2006/32/EC of the European Parliament and of the Council of 5 April 2006 on energy end-use efficiency and energy services and repealing Council Directive 93/76/EEC*. Apr. 27, 2006.

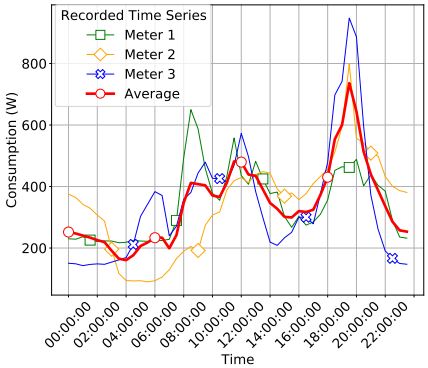
What is sensitive about electric consumption time series?

What is sensitive about a time series?

- Contract type, Devices used
- Socio-economic information[2] (Home presence, Housing space, Number of people at home, Living standards, ...)

What is sensitive about aggregates?

- Extract sensitive attributes from the aggregate.
- Infer the participation of series to the aggregate.
- Knowing that a target is within a series allow to gain the information published alongside the aggregate.



[2] Beckel & al. "Automatic socio-economic classification of households using electricity consumption data". In: *Proceedings of the fourth international conference on Future energy systems*. 2013.

State of the art

- Reconstruction attacks[3]:
  - Find the raw data at the origin of the anonymised data.
  - Requires to build custom queries to build the constraints.
- Membership inference attacks[4,5]:
  - Is a target inside an aggregate?
  - Limited to identify the presence of a single series per attack.
- Attributes inference[6]:
  - Can we detect a sensitive attribute (being at home) from the aggregate ?
  - NILM : **N**on **I**ntrusive **L**oad **M**onitoring
  - Only for individual series and not aggregates.

[3] Cynthia Dwork & al. "The price of privacy and the limits of LP decoding". In: *ACM Symposium on Theory of Computing*. 2007.

[4] Apostolos Pyrgelis & al. "Knock Knock, Who's There? Membership Inference on Aggregate Location Data". In: *NDSS*. 2018.

[5] Luke A. Bauer & al. "Towards Realistic Membership Inferences: The Case of Survey Data". In: *ACSAC*. 2020.

[6] G.W. Hart. "Nonintrusive appliance load monitoring". In: *Proceedings of the IEEE* (1992).

Problem statement

Objective:

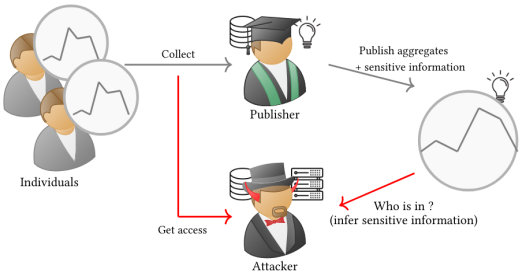
Find the series present in the aggregate to infer information (the sensitive data published alongside the aggregate).

Published data:

- An aggregate (sum of series)
- Metadata about the aggregate: number of series
- Sensitive data published alongside the aggregate (contract type, devices used, ...)

Attacker knowledge:

- Superset of the series present in the aggregate.
- In real life: available to a major provider, disclosed by a data breach.



Is it realistic?

Get access to the data

- Legal access:
  - Actors (other than the publisher) have access to the series of their clients.
- Illegal access:
  - The raw series leaks to the attacker. (1,862 data breach in 2021[7])
- An actor can have all the series of the publication.
- Hard to get, yet not impossible.

Raw time series are not knowledge!

Extract knowledge from raw series requires specific algorithm and labeled ground truth data.

[7] https://iapp.org/news/a/record-number-of-data-breaches-in-2021/.

The SubSum attack

Constraints:

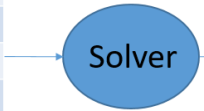
- $\forall t \in \mathcal{T}, A_t = \sum_{\forall i \in \mathcal{S}} S_{i,t} \cdot X_i$
- $\mathcal{S}^{\mathcal{A}} = \sum X_i$
- A_t: Aggregate value at the timestamp t. $S_{i,t}$: Consumption of the individual i at the timestamp t. X_i: Boolean telling whether or not the individual i is in the aggregate. Aggregate size $|\mathcal{S}^{\mathcal{A}}|$

Intuition:

- One constraint per timestamp.
- Each constraint gives a different set of solutions.
  - Solution: Set of individuals whose sum gives the aggregate A_t.
- The true solution is at the intersection of each constraints solutions.
- Ideally, only one solution is found: the real set of series at the origin of the aggregate.

The SubSum attack

	T1	T2	T3	T4
Series 1	1	2	3	1
Series 2	4	3	5	5
Series 3	4	3	4	3
Series 4	3	2	2	4
Series 5	1	3	5	6
Series 6	5	1	4	2
Aggregate (1 + 3)	5	5	7	4



Timestamps	Solutions
$T1$	(1; 2), (1; 3), (1; 6)
$T2$	(1; 2), (1; 3), (1; 5), (2; 5), (3, 4)
$T3$	(1; 3), (1; 6), (2; 4), (4; 5)
$T1 \cap T2$	(1; 2), (1; 3)
$T1 \cap T2 \cap T3$	(1; 3)

Figure: SubSum example

The SubSum attack: experiments

Goal:

- What are the conditions required to make our attack work?
- How long does it take?

Experimental setup:

- Success: Find **all** the existing solutions in the impaired time (time budget: θ).
- Experiments: exploring the parameters space (number of series $|\mathcal{S}|$, aggregate size $|\mathcal{S}^{\mathcal{A}}|$, number of constraints $|\mathcal{A}|$)
- Data: 4500 series (to the W) at 1/2h step over 1,5 years.
  - CER ISSDA[8]& London Households Energy Consumption dataset[9]
- Solver: Gurobi, Computing configuration: 2 cores, 8Gb of RAM.

[7] https://www.ucd.ie/issda/data/commissionforenergyregulationcer/

[9] https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households

Minimal number of constraints to get at least a success

Symmetrical results for aggregate sizes above 50% of the total population.

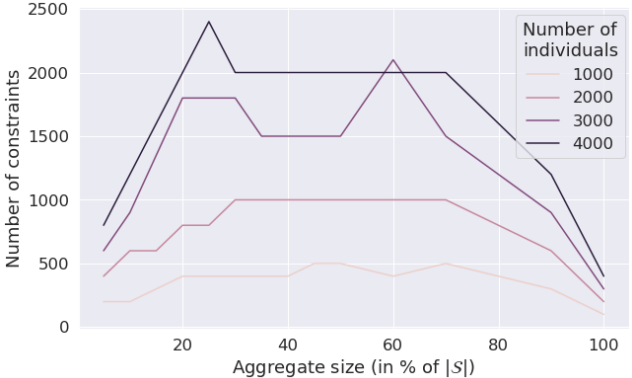


Figure: Y axis: absolute number of constraints. Parameters: Dataset = ISSDA-30m, $|\mathcal{S}| = \{1000, 2000, 3000, 4000\}$, θ=24h, p=2, 20 repetitions

Minimal number of constraints to get at least a success

The number of constraints required for a successful attack is of the same order as aggregate size attacked.

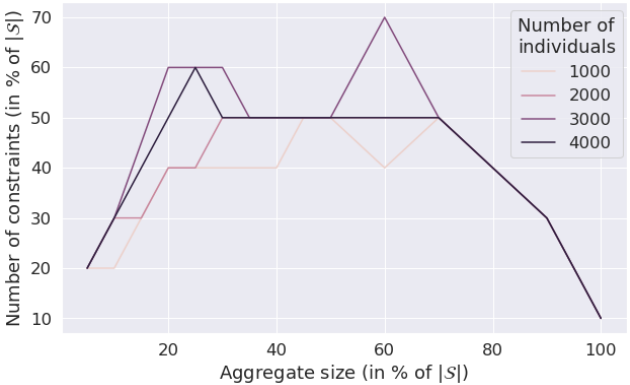
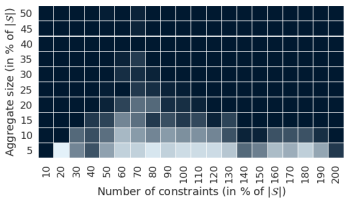


Figure: Y axis: number of constraints in % of the number of individuals. Parameters: Dataset = ISSDA-30m, $|\mathcal{S}| = \{1000, 2000, 3000, 4000\}$, θ=24h, p=2, 20 repetitions.

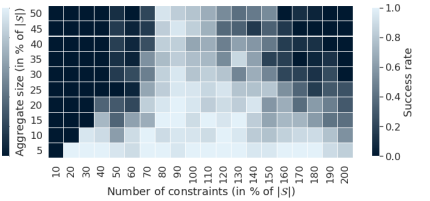
Success depending on the time budget

Higher wall time leads to more success (Except when the number of constraints is too low or too high).

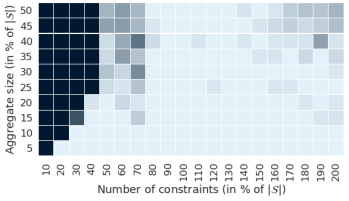
Parameters : dataset = ISSDA-30m, $|\mathcal{S}| = 2000$, $\theta = \{1000\text{s}, 2000\text{s}, 4000\text{s}, 8000\text{s}\}$, $p = 2$, 20 repetitions.



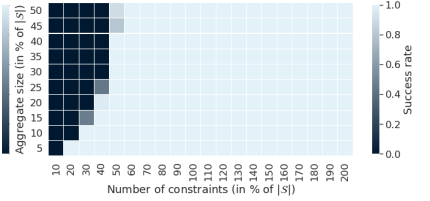
(a) $\theta = 1000$s



(b) $\theta = 2000$s



(c) $\theta = 4000$s



(d) $\theta = 8000$s

Experimentation time depending on the time budget

- When the number of constraints is too low:
  - Attack fails due to wall time hit.
- When the wall time is too high.
  - The attack should be a success but the time increases linearly with the number of constraints.

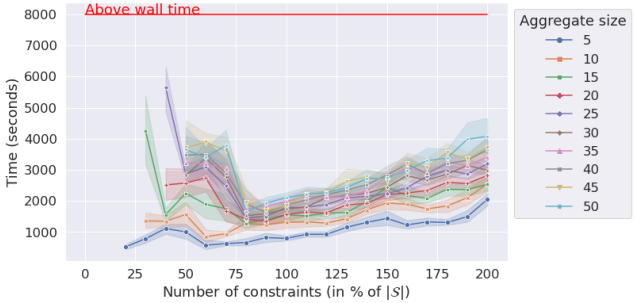


Figure: Parameters : dataset = ISSDA-30m, $|\mathcal{S}| = 2000$, $\theta = 8000$s, $p = 2$, 20 repetitions.

Success for larger datasets

- We are able to attack the whole population available (4500 series).
- Results similar to those obtained with smaller datasets.

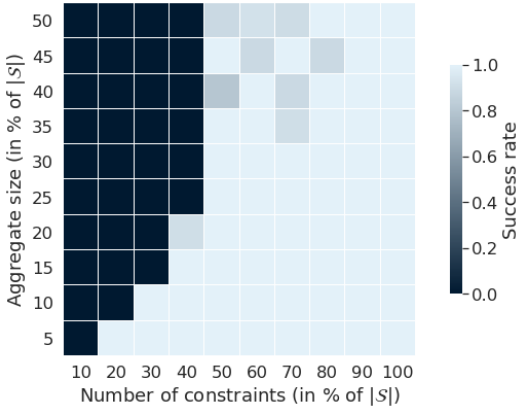


Figure: Parameters: $|\mathcal{S}| = 4500$, $\theta = 24$h, $p = 100$, 20 repetitions.

Number of solutions found

- Exact solution found in almost all runs (98%).
- For the worst-case run, the set of solutions contained three solutions... But:
  - The three solutions only differ on a single time series.
  - Membership inference for all time series except one.

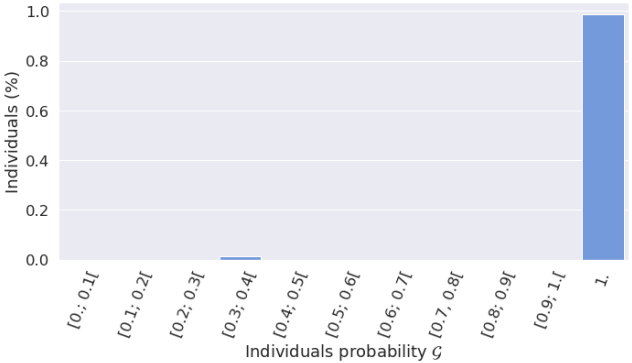


Figure: Parameters: dataset = ISSDA-30m, $|\mathcal{S}| = 4500$, $\theta = 24$h, $p = 100$, $|\mathcal{S}^{\mathcal{A}}| = 225$ (5%), $|\mathcal{A}| = 900$(20%).

Key Takeaways

- Electricity consumption time series.
  - Sensitive data.
  - Published in open data as aggregates (sum / average).
- The SubSum attack:
  - A subset-sum based attack on aggregated time series.
  - Successful attack when the requirements are met.
  - Able to attack aggregates of 2000 series with a month of data in less than 24h.
- Future works:
  - Try the attack on other kind of series.
  - Reduce background knowledge.
  - Cope with noisy aggregates.