Privacy Friendly Aggregation of Smart Meter Readings, Even When Meters Crash

Jaap-Henk Hoepman
TNO / Radboud University
The Netherlands
jaap-henk.hoepman@tno.nl,jhh@cs.ru.nl

ABSTRACT

A well studied privacy problem in the area of smart grids is the question of how to aggregate the sum of a set of smart meter readings in a privacy friendly manner, i.e., in such a way that individual meter readings are not revealed to the adversary. Much less well studied is how to deal with arbitrary meter crashes during such aggregation protocols: current privacy friendly aggregation protocols cannot deal with these type of failures. Such failures do happen in practice, though. We therefore propose two privacy friendly aggregation protocols that tolerate such crash failures, up to a predefined maximum number of smart meters. The basic protocol tolerates meter crashes at the start of each aggregation round only. The full, more complex, protocol tolerates meter crashes at arbitrary moments during an aggregation round. It runs in a constant number of phases, cleverly avoiding the otherwise applicable consensus protocol lower bound.

CCS CONCEPTS

•Security and privacy →Privacy protections; •Computer systems organization →Reliability;

KEYWORDS

Smart grid, privacy friendly aggregation, fault tolerance

ACM Reference format:
Jaap-Henk Hoepman. 2017. Privacy Friendly Aggregation of Smart Meter Readings, Even When Meters Crash. In *Proceedings of The 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids, Pittsburgh, PA USA, April 2017 (CPSR-SG 2017),* 5 pages.
DOI: http://dx.doi.org/10.1145/3055386.3055389

1 INTRODUCTION

Energy infrastructures, especially for electrical power supply, are growing increasingly complex. More and more micro power production units based on solar power or wind energy are connected to the electrical grid. Power consumption is becoming less predictable due to larger variability in life styles and the advance of electric vehicles that need charging. These trends have led to the development of so-called smart grids. Smart grids govern the intelligent automation of the complete transmission and distribution infrastructure that is needed for electric power transport from the energy supplier (generating the energy) all the way down to the end user (consuming the energy). In the future, smart grids will even monitor and control energy consumption of household appliances like dishwashers and the charging of batteries of electric vehicles.

Smart grids are a potential privacy risk [7, 16]. It has been shown that, when sampling power consumption in sufficient detail, smart grid measurements are rich enough such that a measuring trace can easily be de-anonymised with the help of other, publicly available data sources [13]. And as Cavoukian *et al.* argues [5]:

> The inside of a home is the most private of places, and is recognized at the highest judicial levels. (...) Capturing the flow of electricity into one's home, and the manner in which it is used over a period of time, may be revealing and highly intrusive.

In fact, exactly these privacy concerns delayed the introduction of the smart electricity grid in the Netherlands in the first decade of the 21st century.

For this reason smart grid privacy is one of the topics studied in the SEGRID (Security for Smart Electricity GRIDs) project. Within SEGRID Distribution System Operators (DSOs), manufacturers, knowledge institutions and universities collaborate to enhance the protection of smart grids against cyber-attacks (including privacy invasions).

An important functionality of smart grids is the ability to predict energy consumption for (small) parts of the grid, e.g. a block of houses in a city. For this the aggregate energy consumption of all houses in this block must be monitored over a period of time. Privacy friendly solutions to this problem — that allow the DSO to learn the total energy consumption of a group of households without learning the individual energy consumption patterns — have therefore been extensively studied. As becomes clear from the review of the state of the art in section 7, these protocols do not tolerate failures, unfortunately. Not necessarily because such failures would compromise privacy, but rather because these protocols expect *all* meters to provide their inputs. Failure to provide an input either makes the protocol wait forever, or prevents it to actually compute the (partial) aggregated value because some essential data is missing.

We note that such failures do occur in practice, either because of communication failures or because of smart meter malfunctioning. In this paper we therefore study the issue of how to make such

This research is part of SEGRID (Security for Smart Electricity GRIDs) supported by the EU FP7 Programme under Contract No. 607109.
Mon Mar 6 19:14:41 2017 +0100 / 1598e7a / subset-aggregate-reading-cpsr.tex.

CPSR-SG 2017, Pittsburgh, PA USA

978-1-4503-4978-9/17/04...$15.00
DOI: http://dx.doi.org/10.1145/3055386.3055389

privacy friendly aggregation protocols resilient to crash failures. To be precise, let n be the total number of smart meters whose measurements need to be aggregated. We present two protocols that can tolerate t out of n crash failures while still aggregating the total energy consumption of all non-faulty meters in each round. Our contribution is theoretical in nature, showing that both privacy friendliness and fault tolerance can be achieved in the smart grid domain. We therefore did not implement a prototype and also did not perform any real-world performance analysis (or run a simulation to obtain similar results).

The first protocol, presented in section 4, requires crash failures to happen only at the start of each round, and that no failures occur during the aggregation process itself. This makes sense in applications were aggregation happens rather infrequently: in this case meter crashes almost certainly happen in between two aggregation rounds. Section 5 presents a protocol that does not have this restriction and that tolerates crash failures at arbitrary times, at the cost of a two-fold increase in communication complexity. We analyse the privacy properties of our protocols in secion 6. Our conclusions are presented in section 8.

2 MODEL AND PROBLEM STATEMENT

We use a setting similar to Erkin *et al.* [11], except that we allow crash failures to occur. This means the setting is as follows. We assume a fixed number n of smart meters, that each are programmed to report their total energy consumption over a fixed period of time (corresponding to what we will call a round later on). These smart meters are capable of performing basic arithmetic computations, storing temporary results linear in the number of meters n and can directly communicate, point-to-point[1], with each other. We do not assume a separate aggregator or other interconnecting device. We note that this makes our solutions more generally applicable. In particular, if several smart meters are interconnected in a bus-like fashion (similar to an office Ethernet network) this still allows the meters to exchange point-to-point messages. The communication links are assumed to be secure: they guarantee the confidentiality of the messages and the authenticity of both the sender and the receiver. The latter requirement ensures that when meter i sends a message to meter j, it is sure that only meter j learns the contents of the message, and that j is sure it was sent by i.

A threshold of at most t, $0 \leq t < n$ of these meters may fail. Meters fail by crashing [15]: some of its messages may be delivered correctly to their intended recipients, others may fail to arrive, but crashes never introduce spurious, wrong, messages into the system. Our first protocol assumes that such crash failures only happen at the start of a round. Our second lifts that restriction. We note that the model underlying the first protocol makes sense in applications were aggregation happens rather infrequently, in which case meter crashes almost certainly happen in between two aggregation rounds.

We assume a synchronous system (or at least a system in which all messages that are sent are delivered within a fixed and known waiting time), such that all messages that have been sent at the end of a protocol step will have been received at the start of the next step (this step can be delayed until the bounded waiting time guarantees delivery of all messages to all meters).

We do not assume any other active entities such as aggregators or other trusted third parties. Instead our goal is to develop a protocol that allows each smart meter to compute the aggregation of the individual measurements m_i of all non-faulty meters i, while keeping individual measurements private. Our adversarial model is honest-but-curious (cf. [18]): meters behave according to the protocol, but may try to learn more information by overhearing messages exchanged with other meters. Moreover they can collude with at most $n - t - 2$ other meters to recover the individual measurement of a meter not in the colluding set. This is clearly the maximum amount of collusion we can tolerate: if $n - t - 1$ meters collude, and t non-colluding meters fail, then subtracting the individual measurements of the colluding meters from the aggregated sum reveals the measurement of the single remaining honest meter.

3 PRELIMINARIES

3.1 Measurements

Let $\mathbb{Z}_q$ be a field of degree q large enough to contain the aggregate measurement of n meters. We assume each measurement m_i of meter i is an element of $\mathbb{Z}_q$. In fact we assume that m_i is uniformly distributed over the first q/n elements $[0, \ldots, \lfloor q/n \rfloor]$ of $\mathbb{Z}_q$ (or rather, in our security analysis we assume that knowing the sum of two measurements does not allow us to infer anything about any of the two individual measurements). This also ensures that summing all measurements never leads to a wrap around in $\mathbb{Z}_q$.

We will also assume that measurements from a single meter are independent. The sequence of measurements each meter emits consists of random samples from $[0, \ldots, \lfloor q/n \rfloor]$. In practice this is not true: a household consuming a certain amount of electricity now is quite likely consuming a similar amount of energy in a few minutes. However, without this assumption it is impossible to argue that given a few consecutive aggregates from a few smart meters, one cannot derive the individual measurement of some of them. For example, suppose we are given three aggregate values a_0, a_1 and a_2, where we know these aggregates are from meter 0 and 1, meter 1 and 2, and meter 0 and 2 respectively. Suppose the meter readings never change in this timeframe, i.e. $a_0 = m_0 + m_1$, $a_1 = m_1 + m_2$ and $a_2 = m_0 + m_2$. Then $a_0 + (a_1 - a_2) = 2 * m_1$, i.e. revealing meter 1's measurement.

3.2 Secret sharing

Lagrange interpolation allows one to recover an arbitrary point on a polynomial of degree $d - 1$ when given d distinct points on that polynomial.

Definition 3.1 (Lagrange coefficients). For a set $I \subseteq \{1, \ldots, n\}$ and field $\mathbb{Z}_q$ with $q > n$ we define the *Lagrange polynomials* $\lambda_i^I(x)$ as

$$\lambda_i^I(x) = \prod_{t \in I \setminus \{i\}} \frac{x - t}{i - t} \in \mathbb{Z}_q^*[x] \,,$$

[1] In our protocols meters also broadcast messages to all other meters. We do not assume that this broadcast is atomic: if a meter fails a broadcast may deliver the message to a subset of the meters. This means such a broadcast can simply be implemented as sending n point-to-point messages, while using a real, more efficient, broadcast primitive if the infrastructure provides it.

and the *Lagrange coefficients* as $\lambda_i^I = \lambda_i^I(0)$. Then, for any polynomial $P \in \mathbb{Z}_q[x]$ of degree at most $|I| - 1$,

$$P(x) = \sum_{i \in I} \lambda_i^I(x) P(i)$$

and in particular

$$P(0) = \sum_{i \in I} \lambda_i^I P(i) \ .$$

Shamir's d-out-of-n secret sharing is based on this technique [17]. That is, let $s \in \mathbb{Z}_q$ be a secret, pick $\beta_1, \ldots, \beta_{d-1}$ at random from $\mathbb{Z}_q$ and define

$$P(x) = s + \sum_{i=1}^{d-1} \beta_i x^i \ .$$

Then P is a randomly chosen $(d-1)$ degree secret-sharing polynomial over $\mathbb{Z}_q$, such that $P(0) = s$. The n secret shares are given as $s_i = P(i)$. Given an arbitrary set $\{s_{i_1}, \ldots, s_{i_k}\}$ of such shares with $I = \{i_1, \ldots, i_k\}$ the set of indices such that $|I| = k \geq d$, then using the corresponding Lagrange coefficients λ_i^I allows us to recover the secret as $\sum_{i \in I} \lambda_i^I s_i = P(0) = s$.

In our protocol we use the fact that secret sharing is additive.

Lemma 3.2. *Let P and P' be randomly chosen $(d-1)$ degree secret-sharing polynomials over $\mathbb{Z}_q$, where $s = P(0)$ (with secret shares $s_i = P(i)$) and $s' = P'(0)$ (with secret shares $s'_i = P'(i)$). Then $s_i + s'_i$ is a secret share for the secret $s + s'$.*

Proof. We know $\sum_{i \in I} \lambda_i^I s_i = P(0) = s$ and $\sum_{i \in I} \lambda_i^I s'_i = P'(0) = s'$. Then $\sum_{i \in I} \lambda_i^I (s_i + s'_i) = s + s'$ as required. □

4 THE BASIC PROTOCOL

The following protocol aggregates the measurements of all non faulty meters from a set of n meters, when at most t meters are faulty and crash only at the start of each round. Define $d = n - t$, i.e., the number of measurements we are sure to receive in a round[2]. We will deal with crash failures at arbitrary times in the next section.

In each round, the protocol executes the following four synchronous phases. Note that all computations are done in $\mathbb{Z}_q$. For ease of notation we write m_i for the measurement of meter i in the current round (i.e., we do not add another subscript to indicate the round).

- In phase A, each (non-faulty) meter i does the following:
  - Let m_i be the current meter reading to be aggregated.
  - Meter i constructs a random $d-1$ degree polynomial P_i such that $P_i(0) = m_i$.
  - For each $j \in \{1, \ldots, n\}$, i.e., including i, meter i sends secret share $f_{i,j} = P_i(j)$ of m_i to meter j.
- In phase B, each (non-faulty) meter j does the following:
  - Meter j sets local variable $I_j = \emptyset$.
  - It receives $f_{i,j} = P_i(j)$ messages sent to it in phase A by all non-faulty meters i, and adds i to I_j.
- In phase C, each (non-faulty) meter j does the following:
  - Meter j computes the aggregate secret share $F_j = \sum_{i \in I_j} f_{i,j}$, i.e., it sums all received values. In essence this constructs a secret share for the aggregated meter value $\sum_{i \in I_j} m_i$.
  - Meter j broadcasts F_j to all meters $i \in \{1, \ldots, n\}$, i.e., including j.
- In phase D, each (non-faulty) meter i does the following:
  - Meter i sets local variable $J_i = \emptyset$.
  - It receives F_j messages sent to it in phase C by all non-faulty meters j and adds j to J_i .
  - Meter i computes $F = \sum_{j \in J_i} \lambda_j^{J_i} F_j$ and returns this as the aggregated value.

We note the following.

In phase A, each meter sends n secret shares and in phase C each meter broadcasts F_j, so a total of $O(n^2)$ messages are sent and $O(n)$ messages are broadcast.

Because of the assumption that the system is synchronous and that a meter fails only at the start of an aggregation round, in phase B either all meters j receive a share from node i, or no node j does. In other words, all meters j receive values from the same set of meters $I = I_j$. Also, because we assume at most t meters fail at the start of round A, each meter receives at least $n - t = d$ secret shares. In other words, $|I_j| \geq d$. By the same reasoning in phase D we have $J_i = I$ for all non-faulty meters i.

Therefore we have

$$\begin{aligned} F &= \sum_{j \in J_i} \lambda_j^{J_i} F_j = \sum_{j \in I} \lambda_j^I F_j = \sum_{j \in I} \lambda_j^I \sum_{i \in I} f_{i,j} \\ &= \sum_{j \in I} \sum_{i \in I} \lambda_j^I f_{i,j} = \sum_{i \in I} \sum_{j \in I} \lambda_j^I P_i(j) \\ &= \sum_{i \in I} P_i(0) = \sum_{i \in I} m_i \end{aligned}$$

for all non-faulty meters. The last-but-one step follows from the assumption on the number of failures (which implies that $|I| \geq d$) and the construction of the secret-sharing polynomial P_j.

5 DEALING WITH CRASH FAILURES AT ARBITRARY TIMES

The protocol in the previous section only behaves correctly when meters crash at the start of the round, or not at all. The problem lies in the computation of F_j in phase C. When meters can crash at arbitrary times, it is no longer guaranteed that $I_j = I$ for all correct meters j. As a result F_j may contain a share for a meter reading m_z that is not present in other aggregate secret shares F_k. This then creates problems in phase D where it is no longer possible to reconstruct the aggregated meter reading for this round.

One way to solve this is to run a consensus protocol among all nodes after phase B to reach consensus on the set I to compute the aggregate over (making sure that this set is the intersection of all these sets I_j), before computing and broadcasting F_j in phase C. This is expensive though, as such a consensus protocol would use an additional $O(t)$ phases of communication [1] exchanging $O(n(n+t))$ additional messages [2].

There is a more efficient solution out of this dilemma though, if we loosen our requirements a bit, and thus avoid the inefficient lower bounds associated with a full fledged consensus protocol. The idea is to allow each meter to compute the aggregate over a different set of meters, with the restriction that each meter aggregates at least the individual meter readings of the d correct meters. Some meters, however, are allowed to also aggregate meter readings of

[2] Meters also send messages to themselves to simplify the description of the protocol.

meters that crashed somewhere during the round. Note that this is no longer a consensus protocol as we allow correct meters to output different aggregates.

The protocol then runs as follows, proceeding through the following five synchronous phases. Again all computations are done in $\mathbb{Z}_q$.

- In phase A, each meter i does the following:
  - Let m_i be the current meter reading to be aggregated.
  - Meter i constructs a random $d-1$ degree polynomial P_i such that $P_i(0) = m_i$.
  - For each $j \in \{1, \ldots, n\}$, i.e., including i, meter i sends secret share $f_{i,j} = P_i(j)$ of m_i to meter j.
- In phase B, each meter j does the following:
  - Meter j sets local variable $I_j = \emptyset$.
  - It receives and stores $f_{i,j} = P_i(j)$ messages sent to it in phase A by all non-faulty meters i, and adds i to I_j.
  - It broadcasts I_j to all other meters $i \in \{1, \ldots, n\}$, i.e., including j.
- In phase C, each meter i does the following:
  - Meter i sets local variable $R_i = \emptyset$.
  - It receives and stores I_j sent to it in phase B by all non-faulty meters j, and adds j to R_i.
  - It then computes $J_i = \cap_{j \in R_i} I_j$ (i.e., over all I_j received).
  - It broadcasts J_i to all other meters $j \in \{1, \ldots, n\}$, i.e., including i.
- In phase D, each meter j does the following:
  - On receipt of J_i sent to it by meter i in phase C it computes $F_j^i = \sum_{k \in J_j} f_{k,j}$ and sends F_j^i back to i.
- In phase E, each meter i does the following:
  - Meter i sets local variable $K_i = \emptyset$.
  - It receives F_j^i messages sent to it in phase E by all non-faulty meters j and adds j to K_i.
  - Meter i computes $F_i = \sum_{j \in K_i} \lambda_j^{K_i} F_j^i$ and returns this as the aggregated value.

We note the following.

In phase A each meter sends n secret shares, in phase B each meter broadcasts I_j, in phase C each meter broadcasts J_i and in phase D each meter sends n aggregated shares F_j^i. This means the protocol sends $O(2n^2)$ messages and broadcasts $O(2n)$ messages. This makes the protocol twice as expensive (in terms of communication) as the basic protocol.

Because $J_i = \cap_{j \in R_i} I_j$, in phase D we have $J_i \subseteq I_j$. This guarantees that j has all $f_{k,j}$ required to compute F_j^i when it receives J_i from i.

As we are guaranteed that no more than t meters will crash, we receive at least d different F_j^i messages in phase E and thus have $|K_i| \geq d$. This means we receive enough shares to reconstruct the aggregated meter readings.

If a meter k does not crash during the round, then $k \in I_j$ for all non-faulty j in phase B. Therefore, $k \in J_i$ for all non-faulty i in phase C. Hence in phase D, all non-faulty meters j include a share $f_{k,j}$ (for m_k). As a result m_k is aggregated into F_i returned by non-faulty i. In other words, the smart meter readings of all correct meters are certainly aggregated.

We in fact have

$$\begin{aligned} F_i &= \sum_{j \in K_i} \lambda_j^{K_i} F_j^i = \sum_{j \in K_i} \lambda_j^{K_i} \sum_{k \in J_i} f_{k,j} \\ &= \sum_{j \in K_i} \sum_{k \in J_i} \lambda_j^{K_i} f_{k,j} = \sum_{k \in J_i} \sum_{j \in K_i} \lambda_j^{K_i} f_{k,j} = \sum_{k \in J_i} \sum_{j \in K_i} \lambda_j^{K_i} P_k(j) \\ &= \sum_{k \in J_i} P_k(0) = \sum_{k \in J_i} m_k \end{aligned}$$

The last-but-one step follows from the size of K_i ($|K_i| \geq d$) and the degree and construction of the secret sharing polynomial P_k.

6 ANALYSIS

The correctness and message complexity of each of the protocols has already been argued in the sections above. In terms of resources consumed on the smart meters themselves we observe the following. Both protocols use basic arithmetic operations to create secret shares and to combine the values they receive. As this involves computing values for (or processing values from) at most n other meters, each smart meter executes $O(n)$ basic operations. Similarly, each smart meter needs to store $O(n)$ intermediate results. This is practical as long as n is a reasonable value (say the number of households in a street or a block of houses).

What remains to be shown is that the protocols are indeed privacy preserving even when $n - t - 2$ meters are malicious. We will show this for the more advanced protocol from section 5; the argument for the basic protocol is similar.

As argued in section 3 we must study privacy of individual measurements within a single round. So assume without loss of generality that in a single round meter 0 and 1 are honest and fault-free, whereas all other meters are either honest-but-curious and colluding ($n - t - 2 = d - 2$ of them) or fail (t of them). If less meters fail, those meters must be honest as well because of the upper bound of malicious meters. Assume without loss of generality that meters 2 up to $d-1$ are malicious. These meters receive $f_{0,j} = P_0(j)$ and $f_{1,j} = P_1(j)$ (for $j \in \{2, \ldots, d-1\}$) in phase B (i.e., at most $d-2$). They also receive $F_i^j = \sum_{k \in J_i} f_{k,i}$ for $j \in \{2, \ldots, d-1\}$ and all non-faulty i. However, as both 0 and 1 are non faulty, 0 and 1 are member of every J_i and hence both $f_{0,i}$ and $f_{1,i}$ are always summed together in every F_i^j received by the malicious nodes. This gives the malicious meters no information about the distribution of $m_0 = P_0(0)$ (or $m_1 = P_1(0)$) as any possible value for $m_0 = P_0(0)$ (or $m_1 = P_1(0)$) can be explained by a choice of value for $f_{0,i}$ for some honest or failed meter $i \geq d$ (which then fixes $f_{1,i}$ and hence $m_1 = P_1(0)$).

7 STATE OF THE ART

The problem of aggregating the sum of a set of smart meter measurements without learning each individual measurement has been extensively studied [10]. Typically it is assumed that smart meters communicate with an additional central entity called the *aggregator*, and that smart meters cannot communicate directly with one another (see e.g., [4, 12]). Sometimes direct communication between two smart meters is assumed to be possible [14].

The problem studied in this paper is of course an instance of the more abstract problem of securely computing an arbitrary function of some individual inputs, while keeping the inputs themselves

private. The study of these so-called secure multiparty computations was started by the seminal work of Yao [18]. A good recent overview of the state of the art is given by Cramer *et al.* [6]. Even though significant improvement to the complexity of generic secure multiparty computations has made practical implementations a reality [3], tailor made solutions are still more efficient.

Erkin *et al.* [9, 11] started studying the smart meter aggregation problem in a setting where no additional active entities (like a separate aggregator) exist, and instead smart meters can communicate directly with one another[3]. This setting is similar to that of secure multiparty computations. Their protocol requires a one-time setup phase exchanging $O(n^2)$ messages. Each round then uses $O(n)$ broadcast messages, in which a meter broadcasts its own measurement homomorphically encrypted in such a way that these encryptions still can be aggregated to obtain the aggregated sum of all measurements.

The Erkin *et al.* protocol also works in the 'crash-at-start' model to collect the aggregate by simply defining $n' = n - t'$ where t' is the actual number of failures that can be observed when the messages are actually delivered. This no longer works if the random values used in Erkin's protocol are derived from a shared seed exchanged once during setup. In the latter case their $R_{(i,p)}$ contains also the randomness of devices that do not send their values and which therefore does not get canceled out. Their protocols do not tolerate crash failures during the aggregation round.

Efthymiou and Kalogridis [8] deal with failures too, but the privacy guarantees their protocol offers are extremely weak. Meter readings are sent to the aggregator using an 'anonymous' identifier (i.e. without containing a reference to the true identity of the household associated with the meter), yet this identifier stays the same throughout and hence can be used to link different meter readings over time, and in the end re-identify the associated household.

8 CONCLUSIONS

We have presented two protocols that allow individual smart meters to compute the aggregate power consumption of all (non-faulty) meters in the group they belong to in a privacy-preserving fashion, even when an arbitrary subset of the meters may fail by crashing. Our second protocol even tolerates crashes during the aggregation process itself, using only a constant number of message exchange phases for each aggregation round. This is a significant improvement over the state of the art.

In terms of further research we note that our protocols send a quadratic (in the number of smart meters) number of messages. It would be interesting to investigate the possibility to lower the message complexity without sacrificing the fault tolerant properties of our protocol. Related to this question is how to deal with grid setups that do not allow meters to communicate with each other directly, and where a separate aggregating entity is assumed.

[3]Note that this model can be securely simulated by the more traditional model where the smart meters communicate with a central aggregator if that aggregator can be trusted to relay messages between smart meters reliably

REFERENCES

[1] Marcos Kawazoe Aguilera and Sam Toueg. A simple bivalency proof that t-resilient consensus requires $t + 1$ rounds. *Inf. Process. Lett.*, 71(3-4):155–158, 1999.

[2] Eugene S. Amdur, Samuel M. Weber, and Vassos Hadzilacos. On the message complexity of binary byzantine agreement under crash failures. *Distributed Computing*, 5(4):175–186, 1992.

[3] Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas P. Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael I. Schwartzbach, and Tomas Toft. Multiparty computation goes live. *IACR Cryptology ePrint Archive*, 2008:68, 2008.

[4] Claude Castelluccia, Einar Mykletun, and Gene Tsudik. Efficient aggregation of encrypted data in wireless sensor networks. In *2nd Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2005), 17-21 July 2005, San Diego, CA, USA*, pages 109–117. IEEE Computer Society, 2005.

[5] A. Cavoukian, J. Polonetsky, and C. Wolf. Smartprivacy for the smart grid: embedding privacy into the design of electricity conservation. *Identity in the Information Society*, 3(2):275–294, August 2010.

[6] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[7] EDPS. Opinion of the european data protection supervisor on the commission recommendation on preparations for the roll-out of smart metering systems, June 8 2012. European Data Protection Supervisor.

[8] C. Efthymiou and G. Kalogridis. Smart grid privacy via anonymization of smart metering data. In *First IEEE International Conference on Smart Grid Communications*, pages 238–243, 2010.

[9] Zekeriya Erkin. Private data aggregation with groups for smart grids in a dynamic setting using CRT. In *2015 IEEE International Workshop on Information Forensics and Security, WIFS 2015, Roma, Italy, November 16-19, 2015*, pages 1–6. IEEE, 2015.

[10] Zekeriya Erkin, Juan Ramón Troncoso-Pastoriza, Reginald L. Lagendijk, and Fernando Pérez-González. Privacy-preserving data aggregation in smart metering systems: An overview. *IEEE Signal Process. Mag.*, 30(2):75–86, 2013.

[11] Zekeriya Erkin and Gene Tsudik. Private computation of spatial and temporal power consumption with smart meters. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore, June 26-29, 2012. Proceedings*, volume 7341 of *Lecture Notes in Computer Science*, pages 561–577. Springer, 2012.

[12] Flavio D. Garcia and Bart Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In Jorge Cuéllar, Javier Lopez, Gilles Barthe, and Alexander Pretschner, editors, *Security and Trust Management - 6th International Workshop, STM 2010, Athens, Greece, September 23-24, 2010, Revised Selected Papers*, volume 6710 of *Lecture Notes in Computer Science*, pages 226–238. Springer, 2010.

[13] Marek Jawurek, Martin Johns, and Konrad Rieck. Smart metering de-pseudonymization. In Robert H'obbes' Zakon, John P. McDermott, and Michael E. Locasto, editors, *Twenty-Seventh Annual Computer Security Applications Conference, ACSAC 2011, Orlando, FL, USA, 5-9 December 2011*, pages 227–236. ACM, 2011.

[14] Klaus Kursawe, George Danezis, and Markulf Kohlweiss. Privacy-friendly aggregation for the smart-grid. In Simone Fischer-Hübner and Nicholas Hopper, editors, *Privacy Enhancing Technologies - 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings*, volume 6794 of *Lecture Notes in Computer Science*, pages 175–191. Springer, 2011.

[15] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[16] E. L. Quinn. Privacy and the new energy infrastructure. Social Science Research Network (SSRN), February 2009.

[17] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[18] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164. IEEE Computer Society, 1982.