Improving a smart metering system using elliptic curves and removing the trusted dealer

Ricard Garra
Dept. Matemàtica,
Univ. de Lleida.
C. Jaume II, 69, Lleida
garra@matematica.udl.cat

Santi Martínez
Dept. Matemàtica,
Univ. de Lleida.
C. Jaume II, 69, Lleida
santi@matematica.udl.cat

Josep M. Miret
Dept. Matemàtica,
Univ. de Lleida.
C. Jaume II, 69, Lleida
miret@matematica.udl.cat

Francesc Sebé
Dept. Matemàtica,
Univ. de Lleida.
C. Jaume II, 69, Lleida
fsebe@matematica.udl.cat

***Abstract*—Fine-grained electricity consumption information has been proven to allow to infer people's habits from their consumption patterns. Hence, smart metering systems need a secure way to regularly transmit the electricity consumption to the supplier in such a way that the privacy of customers is preserved. Several proposals can be found in the literature. Some of them, classified as aggregative solutions, organize the meters into groups and employ homomorphic cryptography in such a way that the supplier only obtains the aggregated consumption values of the meters in each group.**

In this paper we propose a technique which allows to remove the presence of a trusted dealer in one of the most efficient current proposals. Furthermore, we increase the overall performance of the resulting system by employing elliptic curve cryptography. The resulting system only requires a single message sent from each smart meter to its substation, without needing the meters to communicate among them. With the use of elliptic curve cryptography, the only relevant computation that the smart meters need to perform is a single elliptic curve point multiplication by a scalar. The system has been proven to be feasible and practical even for large neigborhoods including several thousands of meters.

***Index Terms*—cryptography, elliptic curve, privacy, smart meter**

I. Introduction

Smart meters are household devices that transmit information about electricity consumption to the energy supplier in short intervals. Such information allows an accurate prediction of consumption trends so that the production can be properly adjusted avoiding electricity surplus. On the other side, customers can benefit from an accurate billing, reduced prices, and better knowledge of consumption habits. The EU aims to replace at least 80% of electricity meters with smart meters by 2020. Spain is expected to be the first in Europe to replace all analogue electricity meters by the end of 2018.

Unfortunately, smart meters have been proven to be privacy invasive. Fine-grained information about electricity consumption allows to infer sensitive information such as the time a customer leaves or arrives at home, watches TV, or goes to bed. Hence, appropriate solutions have to be deployed in order to preserve customers'privacy.

Privacy-preserving proposals for smart metering can be classified into three main classes:

- *Perturbative:* Meters add some random noise to electricity consumption readings before transmitting them to the energy supplier. In this way, the latter only obtains a noisy version of household consumption profiles. Such solutions [1] require to adjust the noise magnitude for trading-off consumer privacy and data accuracy.
- *Anonymous:* Consumption values are transmitted so that the energy supplier can not determine the identity of transmitters. In proposals [2], [3], each meter has a unique pseudonym which is attached to metering data. In proposal [4] data are transmitted without a pseudonym, while, in [5], each pseudonym is shared among several meters.
- *Aggregative:* Meters are clustered into groups that add their readings prior to transmitting them to their substation. Data are usually aggregated by a trusted party [6] or by making use of some homomorphic cryptosystem [7], [8], [9], [10], [11].

In the smart metering literature it is common to find hybrid proposals combining both perturbative techniques with aggregative ones [8], [10]. In those cases, the meters add some random noise to their data before transmitting them so that the substation obtains a noisy aggregation of consumption readings. The noise added by the meters is usually chosen in such a way that *differential privacy* is provided [12].

A. Aggregative solutions

Aggregative solutions are designed so that the substation can only obtain the addition of meters consumption readings. Most proposals make use of homomorphic encryption.

Proposal [7] requires each meter to have a public key of some additive homomorphic cryptosystem. The transmission protocol has a rather elevated $O(n^2)$ communication cost for a neighborhood composed of n meters.

The approach described in [8] makes use of the Paillier cryptosystem. There exists a neighborhood public key whose private key is distributed among all the meters. The protocol for consumption readings transmission has an efficient linear cost, but the key setup for a distributed Paillier is very complicated when a trusted dealer is not desirable.

The protocol described in [9] uses a distributed threshold ElGamal cryptosystem in order to create a distributed group key y among the meters by themselves. When the meters are requested to send their readings m_i, each of them generates a random integer z_i and encrypts and sends

$$E_y(g^{m_i+z_i}) = (c_i, d_i).$$

The substation then aggregates all messages together, obtaining $(c, d) = (\prod c_i, \prod d_i)$. The value c is sent back to all the meters which employ their private keys x_i to compute a partial decryption

$$T_i = c^{x_i} \cdot g^{z_i}$$

which is transmitted to the substation. The substation then computes

$$D = g^m = d \cdot (\prod T_i)^{-1}.$$

It finally solves an easy instance of the discrete logarithm problem, obtaining the sum of all the readings $m = \log_g D$.

The proposal [10] involves a trusted dealer that generates a set of random values $\{s_0, \ldots, s_n\}$ whose addition is zero. Then, the substation receives s_0 while the remaining values are secretly distributed among the meters. At time step t, each meter M_i, $1 \leq i \leq n$, encrypts its reading m_i by computing

$$c_i = g^{m_i} \cdot H(t)^{s_i},$$

and sends c_i to the substation which aggregates all the received ciphertexts as

$$V = H(t)^{s_0} \prod_{i=1}^{n} c_i,$$

so that the resulting value is $V = g^{\sum m_i}$. Finally, the added readings are obtained by solving the discrete logarithm of V to the base g.

This basic proposal is extended in [10] to provide differential privacy. The mentioned proposal is very lightweight and only requires unidirectional communications from the meters to the substation. Its drawback is the presence of a trusted dealer for key distribution. A similar approach which requires a trusted dealer that generates an RSA modulus is described in [11].

B. Trusted dealer avoidance

The aggregation-based proposal from [10] has been proven to be secure, and has very reduced computational and communication costs. Unfortunately, it requires a trusted dealer for key setup. As mentioned in [13], the requirement of a trusted dealer can be replaced with a TTP-free protocol permitting full-length cleartext homomorphic addition.

The proposals [8], [11] allow homomorphic full-length cleartext addition but require complicated protocols for a distributed generation of hard-to-factor modulus in a trusted dealer-free deployment.

On the other side, the proposal described in [9], which is based on a distributed ElGamal cryptosystem, offers a very simple distributed setup. Its drawback comes from the fact that the resulting aggregated cleartext is obtained after solving a discrete logarithm. Hence, it can only be employed for obtaining the addition of reduced length cleartexts.

C. Contribution and plan of this paper

The objective of this paper is the design of an efficient and easy to configure system able to be employed during the key establishment phase of [10] so as to avoid the presence of a trusted dealer. We propose a method which takes advantage of the simple setup provided by [9] while addressing its limitation of not being able to cope with full-length aggregated cleartexts.

The paper has been structured as follows. Section I has provided an overview to the privacy challenges to be addressed by smart metering systems and has also introduced the objective of our research. Next, our proposal is described in Section II. The security of the proposal is analized in Section III while Section IV proves its feasibility by means of experimental results. Finally, the paper is concluded in Section V.

II. Our proposal

Along the paper we assume the use of elliptic curve cryptography [14] which significantly reduces the size of the involved messages reducing also the computational cost.

The required trusted dealer in [10] will be avoided by employing [9] at the key establishment phase.

The system in [9] allows the transmission of short cleartexts by the smart meters, whose addition is obtained by the substation. We will assume the maximum bitlength of cleartexts to be 13 bits so that, in a neighborhood of up to 128 smart meters, the aggregated message can be obtained by the substation after solving a 20 bits discrete logarithm problem, which took less than half a second in all our experiments.

By using elliptic curve cryptography, the bitlength of the secret values s_i, $0 \leq i \leq n$, to be distributed during the key establishment in [10] can be reduced to 256 bits (down from the original 2048, even obtaining a slightly higher security level). We propose to split such secrets into twenty 13-bits fragments, and transmit the aggregation of those short fragments using [9] in such a way that the substation is able to obtain $\sum_{i=1}^{n} s_i$ by combining the received aggregated fragments without obtaining any information about the individual secrets. By setting s_0 to be $-\sum_{i=1}^{n} s_i$, we get that $\sum_{i=0}^{n} s_i = 0$.

A. Setup

The setup needs to be done before deploying the system.

- Choose an elliptic curve E defined over a prime field $\mathbb{F}_q$, and a point $P \in E(\mathbb{F}_q)$ with prime order p of at least 256 bits.
- Choose a hash function H that returns a point belonging to the subgroup of $E(\mathbb{F}_q)$ generated by P given an input t representing the current time step.

B. Key establishment

The key establishment phase has to be performed before the first execution of the protocol and each time that a smart meter is added or removed from the neighborhood. The secret s_i generated by each smart meter should be similar in size to p.

- Each smart meter M_i, $1 \leq i \leq n$, generates a random secret $s_i < p$ which is then bitwisely split into l chunks s_{ij} of at most 13 bits each such that

  $$s_i = (s_{il} || \ldots || s_{i2} || s_{i1}).$$

  It is required that $13 \cdot l$ is equal or bigger than the size (in bits) of p.
- The meters and the substation execute the protocol in [9] l times (which can be run in parallel). After each execution j, for $1 \leq j \leq l$, the substation obtains $\sum_{i=1}^{n} s_{ij}$. Then the substation computes

  $$s'_0 = \sum_{j=1}^{l} \left(2^{13 \cdot (j-1)} \cdot \sum_{i=1}^{n} s_{ij} \right) = \sum_{i=1}^{n} s_i,$$

  and finally sets its secret s_0 to be $s_0 = -s'_0 \pmod p$. Note that $\sum_{i=0}^{n} s_i = 0 \pmod p$ as required in [10].

C. Consumption transmission

In a smart metering neighborhood, the meters regularly communicate with their substation. At each time step t, every smart meter M_i sends its consumption reading m_i to the substation SSt. The time step may be simply represented as an integer that is incremented by one after each round.

The consumption readings at time step t are transmitted as follows:

- Each meter M_i (storing its secret s_i) transmits its reading m_i by computing an elliptic curve point C_i as
$$C_i = m_i \cdot P + s_i \cdot H(t), \quad (1)$$
which is then sent to the substation.
- The substation receives and aggregates all the points C_i by computing:
$$C = \sum_{i=1}^{n} C_i = m \cdot P + \sum_{i=1}^{n} s_i \cdot H(t),$$
with $m = \sum_{i=1}^{n} m_i$.
- Then, the substation computes:
$$D = C + s_0 \cdot H(t) = m \cdot P + s \cdot H(t) = m \cdot P,$$
since $s = \sum_{i=0}^{n} s_i = 0$.
- Finally, the substation computes the discrete logarithm of D to the base P to get the aggregated readings of all the meters:
$$m = \sum_{i=1}^{n} m_i = \log_P D.$$

Since the computed discrete logarithm is known to be short, it can be efficiently computed by means of Pollard's Lambda algorithm [15].

III. Security analysis

The security of [9], [10] holds on the assumed difficulty of solving the Computational Diffie-Hellman (CDH) problem. Given a multiplicative group G of prime order p with generator g, and given g^a and g^b (a and b are unknown), the computation of g^{ab} is assumed to be hard. The elliptic curve definition of that problem is straightforward by replacing the multiplicative notation with an additive one.

In [9], it is shown that obtaining a partial aggregation of the values sent by the smart meters is as hard as solving the CDH problem. Since we employ [9] during the key establishment phase, no information about the individual secrets s_i is leaked during that phase.

Regarding the transmission of consumption values, we can use [10, Theorem 1] adapted to elliptic curves which proves that the construction in Eq. 1 satisfies aggregator oblivious security (the aggregator is incapable of obtaining any information other that the desired sum of values) in the encrypt-once model.

IV. Experimental results

In this section we test the feasibility of our proposal through experimentation.

The most time-consuming part of the proposal are the computations of the discrete logarithms required both in 'key establishment' and 'consumption transmission'. We have first considered neighborhoods composed of 128 meters, each of them sending a 13 bits number as a cleartext so that the overall aggregated cleartext requires 7 additional bits. Hence, these discrete logarithms have a 20 bits solution.

By using 13 bits, a transmitted cleartext can accomodate the consumption reading of a client during a 30 minutes period. These figures were already proposed in [9].

We used a custom elliptic curve cryptography library and an implementation of Pollard's Lambda algorithm [15] for solving the discrete logarithm problem. Pollard's Lambda is a generic algorithm for solving the discrete logarithm that can be implemented on any cyclic group and, in particular, on the group of points of an elliptic curve as required in our work. Pollard's Lambda algorithm is efficient at solving discrete logarithms when the solution is known to be in a small interval $\{a, \ldots, b\}$. Its time complexity is $O(\sqrt{b-a})$.

The implementation was done in Java (version 1.8.0_92) and executed on a computer with an Intel® Core™ i5-4460 processor running at 3.2 GHz with 16 GB of RAM, using only one of the four cores.

We solved 1000 instances of the problem obtaining an average time of 0.1 seconds per computation. Note that due to the random nature of the algorithm, it can take much more (or much less) time in some instances. In our experiments, the maximum time never surpassed 0.5 seconds.

We also considered the possibility of assuming larger groups of meters (several buildings, entire streets, neighborhoods or towns). Doubling the size of the group requires one additional bit for representing the aggregated consumption. In order to test the feasibility over larger groups, we solved larger instances of the problem.

Meters	Bits	Average	Maximum
128	20	0.10 s	0.48 s
512	22	0.21 s	0.87 s
2048	24	0.42 s	1.48 s
8192	26	0.81 s	4.29 s
32768	28	1.57 s	7.08 s

TABLE I
AVERAGE AND MAXIMUM TIMES FOR SOLVING THE DISCRETE LOGARITHM PROBLEM OVER ELLIPTIC CURVES

Table I shows, for several group sizes, the bitlength of the discrete logarithm to be computed, and the average and maximum running times.

It can be seen that taking 4 times larger groups (which adds 2 bits to the logarithm) doubles the average running time. This is the expected behavior considering the time complexity of Pollard's Lambda algorithm. Due to its random nature, however, the maximum times are more difficult to predict, but our experiments show that they roughly double every two bits. Nevertheless, these extreme times are very unusual.

Our experiments show that larger groups with even 8192 meters are still affordable, since the average time takes less than a second and, in the worst case, it never takes more than 5 seconds. If larger groups were needed, we could employ a more powerful computer, or implement a multicore version of the algorithm.

The cost of the key establishment phase, which needs to be performed each time a smart meter is added or removed

from the group, is mostly determined by the computation of l discrete logarithms, each of size the bitlength of $2^{13} \cdot n$. The cost of the remaining computations is negligible. In our experiments, in the largest considered neighborhood (32768 meters), the execution of a key establishment phase would take around 30 seconds. Note, however, that a smart metering neighborhood is a rather static scenario so that such executions would be quite infrequent.

V. Conclusion

In this paper, we have proposed a system that allows to remove the need for a trusted third party during the key establishment phase of an existing privacy-preserving smart metering system. We also propose to implement it employing elliptic curve cryptography which allows the use of much shorter keys. The system is based on combining two existing proposals, which were already proven to be secure and private.

In the resulting system, the link between electricity consumption values and the customers is broken by homomorphically adding the readings. Since there is no trusted party, even assuming a corrupted substation and some malicious smart meters, the individual readings are still kept private: the only information such a corrupted coalition can obtain is the addition of the readings of the remaining honest meters. The resulting protocol does not require communication among the smart meters, and reading consumptions are transmitted by sending only one message per round. For the meters, the only relevant computation per round is an elliptic curve point multiplication by a full-length scalar. The substation is required to solve an easy instance of the discrete logarithm problem at each round. With an appropriate choice of parameters, this computation has been shown to be solvable in less than one second in a standard computer.

The elliptic curve point multiplication by a scalar performed by the meters, when using projective coordinates and an appropriate curve, can be achieved in general with 12 multiplications, 2 squares, 6 additions, and 1 doubling in the base field $\mathbb{F}_q$, so that only basic modular operations are needed.

Acknowledgments

Research of the authors was supported in part by the European Regional Development Fund of the European Union in the scope of the "Programa Operatiu FEDER de Catalunya 2014–2020" (project COMRDI16-1-0060), and by the Spanish Ministry of Science, Innovation and Universities (project MTM2017-83271-R). The Cryptography & Graphs Research Group of Universitat de Lleida has been recognized as a Consolidated Research Group by Generalitat de Catalunya (2017 SGR 1158).

References

[1] P.V. Barbosa, A. Brito, and H. Almeida: "A technique to provide differential privacy for appliance usage in smart metering", *Information Sciences*, vol. 370-371, pp. 355–367, 2016.

[2] C. Efthymiou and G. Kalogridis: "Smart grid privacy via anonymization of smart metering data", *Proc. of First IEEE International Conference on Smart Grid Communications*, pp. 238–243, 2010.

[3] S. Finster and I. Baumgart: "Pseudonymous smart metering without a trusted third party", *Proc. of 12th IEEE Intl. Conf. on Trust, Security and Privacy in Computing and Communications*, 2013.

[4] R. Petrlic: "A privacy-preserving concept for smart grids", *Sicherh. vernetzten Syst.*, vol. 18, B1-B14, 2010.

[5] M. Stegelmann and D. Kesdogan: "GridPriv: A smart metering architecture offering k-anonymity", *IEEE 11th Intl. Conf. on Trust, Security and Privacy in Computing and Communications*, pp. 419–426, 2012.

[6] J.M. Bohli, C. Sorge, and O. Ugus: "A privacy model for smart metering", *IEEE 1st Intl. Workshop on Smart Grid Communications*, 2010.

[7] F. García and B. Jacobs: "Privacy-friendly energy-metering via homomorphic encryption", *Proc. of 6th Intl. Conf. on Security and Trust Management*, LNCS vol. 6710, pp. 226–238, 2011.

[8] V. Rastogi and S. Nath: "Differentially private aggregation of distributed time-series with transformation and encryption", *Proc. of SIGMOD'10*, pp. 735–746, 2010.

[9] N. Busom, R. Petrlic, F. Sebé, C. Sorge, and M. Valls: "Efficient smart metering based on homomorphic encryption", *Computer Communications*, vol. 82, pp. 95–101, 2016.

[10] E. Shi, R. Chow, T.-H. H. Chan, D. Song, and E. Rieffel: "Privacy-preserving aggregation of time-series data", *Proc. of NDSS' 2011*, The Internet Society, 2011.

[11] M. Joye and B. Libert: "A scalable scheme for privacy-preserving aggregation of time-series data", *Proc. of FC'2013*, Springer–Verlag Berlin Heidelberg, LNCS vol. 7859, pp. 111–125, 2013.

[12] D. Dwork: "Differential privacy: a survey of results", *Proc. of TAMC'2008*, Springer–Verlag Berlin Heidelberg, LNCS vol. 4978, pp. 1–19, 2008.

[13] T. Jung and X. Li: "Collusion-tolerable privacy-preserving sum and product calculation without secure channel", *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 45–57, 2015.

[14] D. Hankerson, A. Menezes, S. Vanstone: Guide to elliptic curve cryptography, Springer, New York, 2004.

[15] J.M. Pollard: "Monte Carlo methods for index computation (*mod p*)", *Mathematics of computation*, vol. 32, no. 143, pp. 918–924, 1978.